

UNIVERSIDADE FEDERAL DO PARANÁ

GUILHERME FERNANDES DE SOUZA MIGUEL

**AVALIAÇÃO DE ESTABILIDADE DE TRANSMISSÕES DE VÍDEO POR  
PROTOCOLO DE TRANSPORTE MULTICAMINHO**

CURITIBA - PR

2017

GUILHERME FERNANDES DE SOUZA MIGUEL

**AVALIAÇÃO DE ESTABILIDADE DE TRANSMISSÕES DE VÍDEO POR  
PROTOCOLO DE TRANSPORTE MULTICAMINHO**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná.

Orientador : Prof. Dr. Eduardo Parente Ribeiro

Co-orientador: Prof. Dr. Carlos Marcelo Pedroso

CURITIBA - PR

2017

---

M635a

Miguel, Guilherme Fernandes de Souza

Avaliação de estabilidade de transmissões de vídeo por protocolo de transporte multicaminho / Guilherme Fernandes de Souza Miguel. – Curitiba, 2017.

76 f. : il. color. ; 30 cm.

Dissertação - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, 2017.

Orientador: Eduardo Parente Ribeiro – Co-orientador: Carlos Marcelo Pedroso,.

Bibliografia: p. 70-75.

1. Protocolo de Transmissão de Controle de Fluxo (Protocolo de rede de computador). 2. Redes de computador – Protocolos. 3. Sistemas de transmissão de dados. I. Universidade Federal do Paraná. II. Ribeiro, Eduardo Parente. III. Pedroso, Carlos Marcelo . IV. Título.

CDD: 004.62

---



UNIVERSIDADE FEDERAL DO PARANÁ  
Programa de Pós-Graduação em Engenharia Elétrica, PPGE  
Setor de Tecnologia

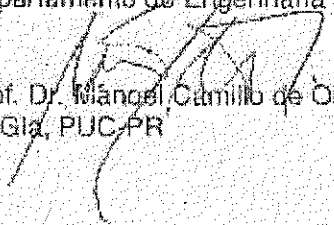
### TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado GUILHERME FERNANDES DE SOUZA MIGUEL intitulada: Avaliação de Estabilidade de Transmissões de Vídeo por Protocolo de transporte Multicaminho, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVADO no rito de defesa. A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 23 de fevereiro de 2017.

  
Prof. Dr. Eduardo Parente Ribeiro  
Departamento de Engenharia Elétrica, UFPR

  
Prof. Dr. Carlos Marcelo Pedrosa  
Departamento de Engenharia Elétrica, UFPR

  
Prof. Dr. Wáncel Camillo de Oliveira Penna Neto  
PPGE, PUC-PR

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Contexto e motivação . . . . .	13
1.2	Objetivos . . . . .	16
1.2.1	Objetivos específicos . . . . .	16
1.3	Artigos relacionados . . . . .	16
1.4	Estrutura da dissertação . . . . .	17
<b>2</b>	<b>CONCEITOS FUNDAMENTAIS</b>	<b>18</b>
2.1	Protocolos de transporte . . . . .	18
2.1.1	O SCTP . . . . .	19
2.1.1.1	Estabelecimento de uma associação SCTP . . . . .	20
2.1.1.2	As variáveis RTT, SRTT e RTO . . . . .	21
2.1.1.3	Pacotes <i>HEARTBEAT</i> . . . . .	23
2.1.1.4	<i>Multihoming</i> . . . . .	23
2.1.1.5	Multistreaming . . . . .	24
2.1.1.6	<i>Slow-Start</i> , controle de fluxo e de congestionamento . . . . .	24
2.1.1.7	Mecanismo de troca de caminho . . . . .	26
2.1.1.8	PR-SCTP . . . . .	28
2.1.1.9	Balanceamento de carga . . . . .	29
2.1.1.10	<i>delay-centric</i> . . . . .	32
2.2	Diversidade de caminhos . . . . .	33
2.2.1	Diversidade de caminhos da internet . . . . .	36

2.3	Medição da qualidade da internet . . . . .	37
2.4	NS2: <i>Network Simulator 2</i> . . . . .	39
2.5	Compressão de vídeo MPEG . . . . .	40
<b>3</b>	<b>ANÁLISE DE DESEMPENHO DE SISTEMAS MULTI-CAMINHO</b>	<b>42</b>
3.1	Implementação no NS2 . . . . .	44
3.2	Métricas de desempenho . . . . .	47
3.2.1	Coleta dos dados . . . . .	47
3.2.2	Variância do atraso . . . . .	49
3.2.3	Índice de Justiça de Jain . . . . .	50
3.2.4	Goodput . . . . .	50
3.3	Cenários utilizados nas simulações . . . . .	51
3.3.1	Cenário 1: Dois caminhos completamente disjuntos . . . . .	52
3.3.2	Cenário 2: Três caminhos completamente disjuntos . . . . .	53
3.3.3	Cenário 3: Três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core . . . . .	53
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>56</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>68</b>
	<b>REFERÊNCIAS</b>	<b>70</b>

## LISTA DE FIGURAS

2.1	Diversidade de Caminhos . . . . .	34
2.2	Diversidade de caminhos na Internet . . . . .	37
2.3	Sequência de formação do GOP . . . . .	41
3.1	<i>Throughput</i> vídeo Aspen vs. tempo sendo a figura (a) GOP(60,3) e a figura (b) GOP(15,2) . . . . .	43
3.2	Gráfico do <i>Throughput</i> TX médio vs. tempo de simulação, dos fluxo ID 1 e 2, com início aleatório entre o primeiro e o último pacote do vídeo. . . . .	44
3.3	Fluxograma de implementação do delay-centric no NS2 . . . . .	46
3.4	Cenários de simulação . . . . .	51
3.5	Cenário 1: Dois caminhos completamente disjuntos . . . . .	52
3.6	Cenário 2: Três caminhos completamente disjuntos . . . . .	53
3.7	Cenário 3: Três caminhos, dois não disjuntos . . . . .	54
4.1	Atraso médio vs. ocupação em transmissão de vídeo para difentes GOPs. . . . .	58
4.2	Goodput médio vs. ocupação em transmissão de vídeo para diferentes GOPs. . . . .	60
4.3	<i>Throughput</i> médio vs. ocupação em transmissão de vídeo com diferentes GOPs . . . . .	62
4.4	Índice de justiça de Jain médio vs. ocupação em transmissão de vídeo com diferentes GOPs . . . . .	63
4.5	Número médio de trocas de caminho vs. ocupação em transmissão de vídeo com diferentes GOPs . . . . .	66

## LISTA DE TABELAS

2.1	Velocidade de <i>Download</i> por operadora . . . . .	38
2.2	Porcentagem de usuários que possuem determinada taxa de transmissão na União Europeia . . . . .	39
2.3	Porcentagem de usuários por tecnologia de acesso na União Europeia . . . . .	39
3.1	Tabela comparativa de característica de GOP . . . . .	42
3.2	Parâmetros Utilizados na API de geração de tráfego . . . . .	45
3.3	Tipos de eventos . . . . .	47
3.4	Exemplo de formato do arquivo de <i>trace</i> . . . . .	48



## LISTA DE SIGLAS

3G	Terceira geração de redes celular
API	<i>Application program interface</i>
CMT	<i>Concurrent Multipath Transfer</i>
CMT-PF	<i>CMT Potentially Failed</i>
DAC	<i>dynamic IP address configuration</i>
DAR	<i>Dynamic Address Reconfiguration</i>
DSL	<i>digital subscriber line</i>
DTC	<i>Discrete Cosine Transform</i>
EC	<i>European Comission</i>
EU	União Europeia
FCC	<i>Federal Communitactions Comission</i>
GOP	<i>Group of Pictures</i>
GOP(M, N)	M representa a quantidade de quadros por GOP e N representa a quantidade de quadros B consecutivos
H.264	MPEG-4 parte 10
HOL	<i>Head of line</i>
IETF	<i>Internet Engineering Task Force</i>
IMS	<i>IP multimedia subsystem</i>
IP	<i>Internet Protocol</i>
IPCC-SCTP	<i>Independent per Path Congestion Control SCTP</i>
IPTV	<i>Television over IP</i>
ITU-T	<i>Internet Engineering Task Force</i>
LS-SCTP	<i>Load-Sharing SCTP</i>
MAC	<i>Media Access Control</i>
MPEG	<i>Moving Picture Experts Group</i>
MSE	Erro médio quadrático
MTU	<i>Maximum Transmission Unit</i>
NGN	<i>New Generation Networks</i>
NS2	<i>Network simulator 2</i>
numStreams	Número total de <i>streams</i>
numUnreliable	Número de <i>streams</i> não confiáveis

PDU	Unidade de dados do SCTP
PoP	<i>Point of Presence</i>
PR-SCTP	<i>Partial Reliability extension</i>
PSTN	<i>Public switched Telephone Networks</i>
quadro B	<i>Bi-directionally</i>
quadro I	<i>Intra</i>
quadro P	<i>Predictive</i>
reliability	Confiabilidade de entrega da mensagem
RFC	<i>Request for comment</i>
RTP	<i>Real Time Protocol</i>
SCTP	<i>Stream Control Transmission Protocol</i>
SFR	<i>Split Fast Retransmit</i>
SIGTRAN	<i>Signaling Transport</i>
SPT	<i>Single Path Transfer</i>
SS7	<i>Signaling System No. 7</i>
TCP	<i>Transmission Control Protocol</i>
TSN	Número de sequência da transmissão
TV	Televisão
UDP	<i>User Datagram Protocol</i>
unordered	Entrega de mensagens fora de ordem
VCEG	<i>Video Coding Experts Group</i>
WAN	<i>Wide Area Network</i>

## LISTA DE SÍMBOLOS

$a\_rwnd$	<i>advertized receiver window size</i>
$A_{c_m}$	Nó de acesso lado Cliente ID 0 a M
$\alpha$	Constante utilizada no cálculo do SRTT
$A_{s_m}$	Nó de acesso lado Servidor ID 0 a M
$\beta$	Constante utilizada no cálculo do SRTT
$C_n$	Nó do cliente ID de 0 a N
COOKIE-ACK	Mensagem de resposta a um pacote COOKIE-ECHO
COOKIE-ECHO	Mensagem de resposta a um pacote INIT-ACK
COOKIE-ECHOED	estado do agente SCTP após o envio de um pacote COOKIE-ECHO
COOKIE-WAIT	estado do agente SCTP após o envio de um pacote INIT
$C_{r_n}$	Nó do core ID 0 a N
Cum-TSN	número de sequência do ponto de transmissão cumulativo
cwnd	janela de controle de congestionamento
$D(P_b, P_a)$	Função diversidade para dois caminhos arbitrários $P_a$ e $P_b$
$D(x)$	Função diversidade
$\bar{d}$	valor médio dos valores armazenados no vetor de atrasos
$d_i$	valor do atraso do pacote na posição $i$ do vetor de atrasos
$f_i$	Flux ID
FID	Número de Identificação de um fluxo
HB	Heartbeat
$HB_{interval}$	Intervalo entre pacotes Heartbeat
$H_i$	RTT do pacote de Heartbeat $i$
$L_{i,j}$	Link entre nó $i$ e nó $j$
Mbps	Megabits por segundo
$n$	Tamanho do vetor de atraso
$N_i$	Nó $i$

$P_k$	Caminho alternativo entre a origem e o destino
$RRT_i$	Medida identificada por I do round trip time
RTO	<i>Retransmission Timeout</i>
$RTO_{ini}$	Valor inicial da variável de Retransmission Timeout
$RTO_{max}$	Valor máximo da variável de Retransmission Timeout
$RTO_{min}$	Valor mínimo da variável de Retransmission Timeout
RTT	<i>Round Trip Time</i>
RTTVAR	Medida de variação do RTT
rwnd	<i>Receiver advertised window</i>
RX	Receptor
SACK	<i>Selective Acknowledgement</i>
$\delta$	Valor aleatório entre -0,5 e 0,5
$S_n$	Nó servidor ID de 0 a N
SRTT	<i>Smoothed Round Trip Time</i>
STATE-COOKIE	Campo da mensagem COOKIE-ECHO
TX	Transmissor
$v_i$	<i>Throughput</i> médio para o fluxo f

## RESUMO

A evolução tecnológica de dispositivos móveis e o aumento da disponibilidade de redes IP, criam novas oportunidades para distribuição de conteúdo *multimedia* pela Internet. Atualmente os usuários têm a possibilidade de acessar filmes, peças de teatros, concertos de música transmitidos ao vivo ou armazenados sob-demanda. Aplicações de tempo-real requerem alta qualidade de serviço, não sendo tolerantes a variações do atraso médio por exemplo. Neste contexto, o *delay-centric* foi proposto para realizar *handover* entre redes móveis no protocolo SCTP. Sua proposta de realizar a troca de caminho utilizando o valor do atraso médio suavizado mostrou uma melhora significativa quando utilizado em aplicações de tempo-real. Vários autores tem estudado a utilização do SCTP padrão, do CMT e do CMT-PF em transmissões de vídeo de tempo real. Este trabalho se propôs a estudar a estabilidade de transmissões de vídeo HD com diferentes características de GOP, por protocolos de transporte multicaminho. Simulações foram realizadas com traçados de vídeo de características distintas e foram analisados, para fins de comparação, o comportamento do *delay-centric* com diferentes histereses, SCTP padrão, CMT e do CMT-PF. Foram utilizados cenários com vários níveis de diversidade e observadas métricas de atraso médio, *Goodput*, justiça na divisão do throughput e *throughput* médio nas interfaces dos agentes que enviavam e recebiam o vídeo. Nos cenários com dois e três caminhos completamente disjuntos, os resultados obtidos confirmaram os dados contidos na literatura onde o *delay-centric*, independente da histerese, teve desempenho igual ou superior do CMT e ao CMT-PF. O SCTP Padrão não se beneficiou da existência de múltiplos caminhos, obtendo o pior desempenho dentre todos os protocolos estudados nos cenários com dois e três caminhos completamente disjuntos. Entretanto, no cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core, os resultados indicaram que o *delay-centric* sem histerese obteve o melhor desempenho dentre todos os protocolos estudados e o SCTP Padrão mostrou melhor desempenho que o CMT, CMT-PF e *delay-centric* com histerese de 10ms, 20ms e 30ms.

Palavras-chave: SCTP. Delay-centric. *Multihoming*.

## ABSTRACT

The mobile devices technological evolution and the increased availability of IP networks, create new opportunities for multimedia content distribution over Internet. Nowadays, users have access to movies, plays or music concerts live over demand via video streaming over the Internet. Real-time applications require high quality service and are not tolerant to delay variations. The delay-centric was initially proposed to make the handover between mobile networks in SCTP by using the smoothed round trip time and delivering a higher user experience on real-time applications. Other authors have studied the use of CMT and CMT-PF in real-time video streaming. The main objective of this work is to study the stability of HD video streaming, with different GOP characteristics, using multipath transport protocols. Video traces were used in the simulations with various path configurations to compare the behavior of delay-centric with different hysteresis, standard SCTP, CMT and CMT-PF. Results were analyzed using the metrics of delay, jitter, Goodput, mean throughput and throughput division justice. In scenarios with two or tree disjoint paths, the results confirmed the previous results on the literature that the delay-centric, independent of hysteresis level, have same or better performance when compared to CMT and CMT-PF. The standard SCTP do not take advantage of the multiple disjoint paths, having the worst performance when compared to the other protocols. In scenarios with tree access nodes connected to two cores, with two access networks sharing the same core, the results showed that the delay-centric with no hysteresis 5have the best performance in comparison to the other protocols, followed by the standard SCTP that presented better performance than CMT, CMT-PF and delay-centric with 10ms, 20ms, 30ms of hysteresis respectively.

Key words: SCTP. Delay-centric. *Multihoming*.

## CAPÍTULO 1

### INTRODUÇÃO

#### 1.1 Contexto e motivação

No início da década de 1970, durante o surgimento das redes baseadas no *Internet Protocol* (IP), redes de telefonia e de comunicação de dados eram vistas como entidades diferentes. Seus requisitos operacionais, escolhas de projeto técnico e padrões de procedimentos eram alguns dos fatores que contribuíam para esta visão. As redes IP tem como característica de projeto a resiliência<sup>1</sup> em casos de falhas [2]. Sua robustez<sup>2</sup> em larga escala se deve, em grande parte à natureza distribuída dos seus protocolos operacionais [4]. Por estas características das redes IP, em menos de uma década e meia, iniciou-se a crescente convergência das redes de comunicação de dados e telefonia [5].

Redes sem fio se tornaram comuns e as pessoas passaram a cada vez mais utilizá-las nas suas atividades diárias. Muitos dos atuais dispositivos possuem múltiplas interfaces de conexão com a internet, mas estes estão impossibilitados de se beneficiar simultaneamente delas por conta de limitações do protocolo IP. O avanço dos dispositivos móveis e das interfaces de rede possibilitou ainda o desenvolvimento de serviços multimídia e de comunicação em tempo real [6]. As redes sociais têm contribuído para essa nova tendência, dando ao usuário a possibilidade de acessar conteúdos sob-demanda ou em tempo-real na Internet. Além das redes sociais, várias empresas que produziam conteúdo áudio-visual para o cinema e TV estão disponibilizando seus conteúdos através de serviços de transmissão de vídeos de alta qualidade pela internet. Teatros e salas de concerto possuem transmissões ao vivo de seus eventos, além de disponibilizar eventos anteriores para seus assinantes [7].

Os protocolos de camada de transporte mais utilizados atualmente são o *Transmission Control Protocol* (TCP) [8] e o *User Datagram Protocol* (UDP) [9]. O UDP é geralmente utili-

---

<sup>1</sup>Conforme definido por Sterbenz et al. [1], Resiliência é “a habilidade de uma rede prover e manter níveis aceitáveis de serviço em face de várias falhas e desafios normais de operação”

<sup>2</sup>Conforme definido por Meyer [3], Robustez é “a habilidade de um sistema em manter suas características quando submetidos a um conjunto de perturbações internas ou externas”

zado em aplicações de tempo real juntamente com o *Real Time Transport Protocol* (RTP). O RTP é um protocolo de camada de aplicação que implementa serviços como: identificação de *payload*, número de sequência, *timestamping* e monitoramento de entrega de pacotes através de relatório RTPC, mas não possui nenhuma implementação de controle de fluxo ou de congestionamento [10]. A inexistência de algoritmos de controle de congestionamento no UDP, ocasiona injustiça no balanceamento de carga dos diversos fluxos de rede. Enquanto o TCP reduz a taxa de envio de pacotes para minimizar o congestionamento de rede, a falta de conhecimento do *status* da rede pelo UDP faz com que ele continue enviando pacotes [8].

Existem fragilidades na performance de qualquer aplicação que utiliza a internet. Perturbações no estado operacional da rede, invisíveis ao usuário final, resultam na degradação da qualidade de serviço oferecida. A existência de congestionamento na rede inviabiliza a utilização de aplicações de tempo real, dadas as suas limitações quanto ao atraso, *jitter* e perda de pacotes. O atraso é o tempo absoluto que um pacote leva para ser transmitido de uma rede de origem a uma rede de destino [11] e o *jitter* é uma medida da variação do atraso. A perda de pacotes pode ser causada por problemas no meio físico ou por descarte de pacotes nos elementos de rede antes de chegarem ao destino final [12].

Em função do aumento da diversidade dos meios de acesso à internet, surgiu a possibilidade de incorporar múltiplas interfaces de rede em um único dispositivo. Isto possibilita o acesso à internet por mais de um caminho, chamado *Multihoming*. Do ponto de vista da camada de transporte, cada interface apresenta um caminho diferente para acesso à rede. Através de mecanismos de monitoramento e de troca de caminhos, é possível aumentar a resiliência e estabilidade de uma conexão, garantindo a continuidade dos serviços mesmo no caso do caminho principal se tornar indisponível. Vários autores abordam a utilização do *Multihoming* combinado com controle inteligente de rotas e redes de circuitos lógicos na tentativa de aumentar a redundância e performance. Entretanto, a efetividade destas abordagens depende da diversidade natural dos caminhos entre destinos finais em termos de *links* físicos, infraestrutura de roteamento, controle administrativo e distribuição geográfica. Se o tráfego entre dois *hosts* não for roteado entre caminhos completamente disjuntos, congestionamento ou falhas de um único *link* ou roteador compartilhado podem afetar a performance ou a disponibilidade destes caminhos [13]. Novas abordagens para remediar possíveis problemas na rede em um contexto fim-a-fim foram propostas, uma vez que os *hosts* de origem e/ou destino



são tipicamente os primeiros a tomarem conhecimento na ocorrência de uma interrupção de um serviço [6].

O *Stream Control Transmission Protocol (SCTP)* foi inicialmente proposto para realizar o transporte de mensagens de sinalização para o *Signaling System No. 7 (SS7)*, sinalização encontrada na tradicional *Public Switched Telephone Networks (PSTN)*, através da rede IP. A primeira especificação do SCTP foi publicada em outubro de 2000 pelo *Signaling Transport Working Group (SIGTRAN)* do *Internet Engineering Task Force (IETF)*, sendo a última revisão publicada na RFC 4960. O SCTP tem se mostrado apropriado para transporte de dados de aplicações tradicionais por prover conexão em associações *multihoming*, dissociação entre a entrega confiável e entrega ordenada de mensagens, e múltiplas *streams* lógicas multiplexadas em uma mesma conexão física. A ocorrência de congestionamento temporário ou falha permanente no caminho primário, ocasiona troca de caminho do fluxo de dados para um dos caminhos secundários. Outras abordagens para a troca de caminho foram propostas, considerando mobilidade, redundância, transmissão simultânea em múltiplos caminhos e troca de caminho baseado em atraso [5].

A ideia de selecionar o caminho primário, utilizando como critério o menor valor do atraso médio *Smoothed Round Trip Time (SRTT)* foi inicialmente proposta por Kelly et al. [14], chamando-a de *delay-centric*. Na época os autores estudaram a utilização do *delay-centric* para realizar o *handover* entre redes móveis de terceira geração (3G) e *Wireless Local Area Networks (WLAN)*, que ainda são muito utilizadas nos dias atuais por telefones móveis. Os experimentos realizados pelos autores mostraram melhorias na qualidade na transmissão de dados para aplicações sensíveis a atraso. Os autores identificaram também que este mecanismo pode prever falha no caminho primário, enviando os dados por um dos caminhos secundários.

Por conta da possibilidade de transmissão de dados por múltiplas interfaces do SCTP, vários autores tem estudado a sua utilização para transmissão de vídeo *streaming* pela internet. Entretanto estes estudos possuem limitações em suas abordagens do mecanismo de troca e da diversidade de caminhos, qualidade da compactação do vídeo e taxa de transmissão das interfaces de rede.

## 1.2 Objetivos

Este trabalho tem como objetivo verificar a estabilidade e desempenho de Transmissões multi-caminho em situações com competição de fluxos de vídeo de alta definição no padrão H.264 em diferentes níveis de diversidade de caminhos, comparando diferentes estratégias de transmissão.

### 1.2.1 Objetivos específicos

- Mensurar o balanceamento do tráfego entre os múltiplos agentes e caminhos.
- Averiguar a justiça na divisão de recursos entre os diversos caminhos.
- Mensurar o comportamento do atraso, jitter, perda de pacotes e throughput

## 1.3 Artigos relacionados

A utilização de sistemas *multihoming* para aumentar a performance e resiliência de comunicações é objeto de estudo de vários autores. A utilização do protocolo *Multipath Transmission Control Protocol* (MTCP) em dados reais de voz foi estudado por Tran et al. [15]. Neste trabalho o autor analisou as principais características de conexões MTCP em conjuntos de dados coletados entre 25 de novembro de 2014 e 29 de abril de 2015, variando em termos de duração e quantidade de dados trafegados. Com esses dados, os autores mostraram que aproximadamente 86% das conexões MTCP analisadas possuíam mais de um fluxo criado imediatamente após o estabelecimento da conexão. Nas conexões remanescentes, os fluxos adicionais foram criados 1 segundo (s) após o estabelecimento da conexão, em cenários onde ocorreram *handover*.

Hussein, Hassan e Landolsi [16] avaliou a performance de transmissões de vídeo em ambientes *multihoming*, avaliando a utilização de múltiplas interfaces e políticas de priorização de determinados tipos de *frames* H.264. Os nós de roteamento foram os responsáveis por enviar os pacotes pelo caminho com menor nível de congestionamento e foram testados os níveis de diversidade de 3 ou 4 caminhos disjuntos, comparando com o sistemas com apenas um caminho e um caminho com prioridade de determinados tipos de pacote. Os autores

observaram que o melhor cenário para as transmissões de vídeo é a transmissão simultânea por 3 ou 4 caminhos.

Ainda analisando a performance de distribuição de vídeos em *multihomed hosts*, Xu et al. [7], analisou a performance da distribuição de conteúdos *multimedia* através de redes sem fio, utilizando duas abordagens do protocolo SCTP *Single Path Transfer* (SPT), que realiza a transmissão de dados em apenas um dos caminhos disponíveis, e o CMT (*Concurrent Multipath Transfer*), que realiza a transmissão de dados por todos os caminhos disponíveis. Como métrica de desempenho, foi utilizada a qualidade percebida pelo usuário, em ambientes com diferentes taxas de transmissão, atraso e perda de pacotes. Os resultados obtidos mostraram melhora da qualidade percebida quando utilizado o CMT nas transmissões de vídeo.

#### **1.4 Estrutura da dissertação**

Após este capítulo introdutório, esta dissertação está organizada da seguinte forma. No capítulo 2 são apresentados os fundamentos sobre o STCP e suas abordagens para realizar a troca de caminhos, diversidade de caminhos e a diversidade de caminhos nas internet, o algoritmo *delay-centric* e sua implementação no simulador de eventos discretos NS2, compressão de vídeos em H.264 e os trabalhos relacionados.

## CAPÍTULO 2

### CONCEITOS FUNDAMENTAIS

#### 2.1 Protocolos de transporte

Protocolos da camada de transporte são os responsáveis por criar conexões virtuais entre processos hospedados em diferentes sistemas finais, tornando possível a troca de mensagens entre estes processos. Os principais protocolos de transporte atualmente utilizados são o UDP e o TCP, especificados nas RFC768 [9] e RFC793 [17] respectivamente.

O UDP é um protocolo de transporte simples, orientado à transação que apenas provê entrega de dados entre processos que rodam em diferentes destinos finais, não garantindo a entrega, ordem ou duplicidade das mensagens.

O TCP é um protocolo de transporte orientado a conexão, concebido para ser parte de uma arquitetura de camada de protocolos, que provê comunicação confiável e ordenada entre pares de processos que são executados em *hosts* de redes de comunicação de dados distintas interconectadas. Ele pode operar por um grande espectro de sistemas de comunicação, de *hosts* diretamente conectados à redes de comutação de circuitos ou de comutação de pacotes. Ele também provê controle de congestionamento, permitindo que conexões TCP trafegando por um enlace de rede congestionado compartilhem de maneira igualitária o enlace.

No final do ano de 1998 o IETF criou o grupo de trabalho SIGTRAN, com o objetivo de “propor uma solução para o transporte de pacotes de sinalização da rede PSTN em redes IP, levando em consideração seus requisitos funcionais e de performance” [18]. Como fruto deste trabalho foi proposto protocolo SCTP, publicado na RFC 4960 [5], tornando-se protocolo de transporte padrão das redes *New Generation Networks* (NGN) e *IP Multimedia Subsystems* IMS. Desde a fase inicial de especificação do SCTP, seus projetistas já previam que este fosse também utilizado para o transporte de qualquer tipo de dados, porém mais de 10 anos se passaram desde a primeira especificação do SCTP e este ainda não foi implementado por padrão em alguns dos sistemas operacionais mais utilizados, tais como *MS Windows* e *Mac*

OSX [5].

### 2.1.1 O SCTP

Assim como o TCP, o SCTP oferece uma conexão confiável, *full-duplex* com mecanismos de controle de fluxo e congestionamento, sendo o termo associação utilizado para descrever a conexão entre os dois sistemas finais. Uma associação tem um significado mais abrangente que o de uma conexão TCP, pois ela pode ser estabelecida utilizando múltiplos endereços IP em cada um dos sistemas finais.

O SCTP implementa novas formas de envio de dados assim como outras funcionalidades e serviços. Algumas das principais funcionalidades e aplicações do SCTP são:

- *Multihoming*: O SCTP possibilita que uma associação seja estabelecida, utilizando múltiplos endereços IP em cada um dos sistemas finais. Essa funcionalidade, chamada *multihoming*, foi inicialmente desenvolvida para aumentar a disponibilidade através da troca para um caminho secundário em caso de falhas no caminho primário. Outras abordagens para a utilização do *multihoming* foram propostas, como por exemplo o balanceamento de carga entre diferentes caminhos e o mecanismo de escolha de caminho baseado no atraso médio suavizado dos pacotes. O *multihoming*, o balanceamento de carga e o mecanismo de escolha de caminho baseado no atraso médio suavizado são descritos nas sessões 2.1.1.4, 2.1.1.9, e 2.1.1.10 respectivamente.
- *Multistreaming*: Permite que uma associação possua múltiplas *streams* lógicas com números de sequência independentes. A utilização do *multistreaming* mitiga a ocorrência do bloqueio de *Head of Line* (HOL), já que a perda de um pacote de uma das *streams* não interfere no envio de mensagens pelas demais, diferente do que acontece no TCP e no MTCP. Essa funcionalidade é descrita na sessão 2.1.1.5.
- *Load Sharing*: É a utilização do *Multihoming* para transferência de dados simultânea em múltiplos caminhos com balanceamento de carga entre eles. Isto permite a aplicações que geram *Bulk Traffic*, caracterizado por dados com tamanhos muito grande, possam escoar de forma mais rápida, beneficiando-se de uma maior taxa de transferência, entre os *hosts* de origem e destino. Esta funcionalidade é descrita na sessão 2.1.1.9.

- *Dynamic Address Reconfiguration (DAR)*: É uma importante extensão do SCTP no contexto do *multihoming* que possibilita a reconfiguração dinâmica de endereços. Ela possibilita a inclusão ou remoção de endereços IP ou solicitar a troca de um caminho primário em uma associação SCTP ativa. A funcionalidade de *multihoming* junto com o DAR podem ser utilizados para prover soluções de *handover* através da camada de transporte. Ela está definida na RFC 5061 [19].
- *SCTP Partial Reliability Extension (PR-SCTP)*: Esta extensão oferece diferentes níveis de confiabilidade e ordenação das mensagens. Estes diferentes níveis de confiabilidade, possibilitam uma maior flexibilidade na entrega de dados, entre o *host* de origem e destino, pela camada de transporte. Essa funcionalidade é descrita na sessão 2.1.1.8. Utilizando as funcionalidades de *Multihoming* e PR-SCTP, o SCTP pode ser utilizado para transferência de dados de aplicações *multimedia*.
- *Controle de congestionamento*: O controle de congestionamento é uma questão central para qualquer protocolo de transporte. Para regular a taxa de envio de dados o SCTP utiliza os mecanismos já conhecidos do TCP de janela de congestionamento *Congestion Window* (*cwnd*) e *slow-start threshold* (*ssthresh*) [5]. Essa funcionalidade é descrita na sessão 2.1.1.6.

### 2.1.1.1 Estabelecimento de uma associação SCTP

A unidade de dados do SCTP (PDU), chamado de pacote SCTP, consiste de um cabeçalho comum e um ou mais *chunks*. Um *chunk* carrega informações específicas que podem ser de controle ou dados do usuário. Uma associação SCTP é iniciada com a troca dos *chunks*: INIT, INIT-ACK, COOKIE-ECHO e COOKIE-ACK. Este processo é chamado de *handshake* de quatro vias e é similar ao descrito por Karn e Simpson [20], provendo proteção contra ataques de sincronização e do tipo negação de serviço. Esta é uma das diferenças entre o SCTP e o TCP, que utiliza o *handshake* de três vias.

O processo de inicialização consiste dos seguintes passos:

1. Inicialmente o agente SCTP de origem envia um *chunk* INIT para o agente SCTP de destino. Esta mensagem deve conter uma *tag* de verificação, composta por um número

randômico entre 1 e 4294967295. Após enviado o INIT, o agente SCTP de origem inicia um *timer* e entra no estado COOKIE-WAIT.

2. O agente SCTP de destino deve responder, imediatamente após o recebimento do INIT, com uma mensagem INIT ACK. O endereço IP de destino deve ser o endereço de origem informado na mensagem INIT recebida. A resposta também deve incluir uma *tag* própria do agente SCTP de destino, além da contida na mensagem recebida. O INIT ACK deve incluir ainda o campo STATE COOKIE contendo: o endereço *Media Access Control* (MAC), o *timestamp* em que foi gerado o STATE COOKIE e seu tempo de vida.
3. Ao receber o INIT ACK do agente SCTP de destino, o *host* de origem deve desativar o *timer* e sair do estado COOKIE-WAIT. Em resposta à mensagem recebida, ele deve enviar o STATE COOKIE recebido no INIT ACK em uma mensagem COOKIE ECHO, iniciar novamente o timer e entrar no estado COOKIE-ECHOED.
4. Quando receber o COOKIE ECHO, o agente SCTP de destino deve responder com uma mensagem COOKIE ACK antes de montar o *Transmission Control Block* e ir para o estado ESTABLISHED.
5. No recebimento do COOKIE ACK, o agente SCTP de origem deve ir para o estado ESTABLISHED, parando o *timer*. Ele deve ainda notificar a camada de aplicação sobre o estabelecimento da conexão.

Os endereços IP das interfaces são enviados e verificados no início da associação e cada endereço de destino é considerado um caminho diferente entre os *hosts*. Ainda no início da conexão, um dos caminhos é designado como caminho primário e os demais como caminhos secundários ou *backup paths*. [5].

#### 2.1.1.2 As variáveis RTT, SRTT e RTO

Em uma comunicação com múltiplos agentes, cada um dos destinos-finais deve conter um contador de erros para cada um dos caminhos. A cada estouro de temporização *Retransmission Timeout* (RTO) em qualquer dos endereços, ou quando um pacote de *HEARTBEAT* de um caminho inativo não for reconhecido, o contador de *timeout* deve ser incrementado. [5].

Cada caminho possui seu próprio RTO. O RTO determina o tempo máximo que o SCTP aguarda pela confirmação de recebimento de um pacote antes de retransmití-lo. Este não pode ser inferior a  $RTO_{min}$ , padrão: 1s e nem superior a  $RTO_{max}$ , padrão: 60s [5].

No início de uma associação o RTO é inicializado com o valor de  $RTO_{ini}$  (padrão: 3s). O RTO é atualizado cada vez que uma medida de tempo de ida e volta *Round Trip Time* (RTT) é realizada para o caminho em questão. O RTT é a medida de tempo entre o instante em que um pacote é enviado até o instante em que o ACK correspondente é recebido. Já nos caminhos secundários isso se dá através dos pacotes *HEARTBEAT* e seus ACKs [5].

Conforme descrito por [21], as variáveis SRTT (RTT suavizado) e RTTVAR (medida de variação do RTT) devem ser atualizadas a cada medição do RTT. Na primeira medida de  $RTT_1$ , os valores das variáveis SRTT, RTTVAR e RTO são atualizados da seguinte forma:

$$SRTT = RTT_1 \quad (2.1)$$

$$RTTVAR = \frac{RTT_1}{2} \quad (2.2)$$

$$RTO = SRTT + 4 \cdot RTTVAR \quad (2.3)$$

A partir da segunda medida de RTT, a cada nova medida  $RTT_i$ , as variáveis SRTT, RTTVAR e RTO são atualizadas como:

$$SRTT_i = (1 - \alpha) \cdot SRTT_{i-1} + \alpha \cdot RTT_i \quad (2.4)$$

$$RTTVAR_i = (1 - \beta) \cdot RTTVAR_{i-1} + \beta \cdot |SRTT_{i-1} - RTT_i| \quad (2.5)$$

$$RTO = SRTT_i + 4 \cdot RTTVAR_i \quad (2.6)$$

Onde  $\alpha$  e  $\beta$  são constantes cujos valores recomendados são: 1/4 e 1/8 [21].



### 2.1.1.3 Pacotes *HEARTBEAT*

Para atualizar o estado (ativo ou inativo) de cada um dos caminhos, o SCTP deve avaliar cada um dos caminhos, certificando sua conectividade. No caminho primário isso se dá através dos pacotes de ACK. Já nos caminhos secundários, essa avaliação é realizada através de pacotes *HEARTBEAT*. A resposta a esses pacotes, *HEARTBEAT ACK*, indica que o caminho se encontra ativo. O RTT de pacotes *HEARTBEAT* ( $H_i$ ), é calculado da seguinte forma:

$$H_i = RTO_i + HB_{interval} \cdot (1 + \delta) \quad (2.7)$$

Onde  $RTO_i$  é o valor atual do RTO para o caminho,  $\delta$  é um valor aleatório entre  $-0,5$  e  $0,5$  e  $HB_{interval}$  o intervalo médio entre pacotes  $H_i$ , cujo valor padrão é 60 s [22].

O valor de  $RTO_i$  tem um efeito mínimo então  $\delta$  é utilizado a fim de introduzir alguma variância na temporização dos pacotes *HEARTBEAT*. De acordo com a equação 2.7 acima, um pacote de *HEARTBEAT* é enviado ao destino, em média a cada  $H_i$  segundos. Se for recebido um *HEARTBEAT ACK* do destino, o caminho é avaliado como ativo. Caso não seja recebida nenhuma resposta, o caminho é marcado como inativo e um novo *HEARTBEAT* é enviado [14].

### 2.1.1.4 *Multihoming*

O suporte à *multihoming* do SCTP permite que uma única associação tenha múltiplos endereços IPs de origem e destino entre dois destinos finais, não requerendo a presença de múltiplas interfaces de rede. Uma interface configurada com mais de um endereço IP já satisfaz este requisito.

Alguns dos endereços de transporte, de um dos participantes da comunicação, pode se tornar inativo por conta de algum erro gerado por uma falha na rede ou por ajustes realizados pela aplicação. Quando existem dados a serem transmitidos e o caminho primário se tornar inativo ou a aplicação explicitamente requisitar o envio de dados para um endereço de transporte inativo, antes de reportar um erro para a aplicação, o SCTP deve tentar enviar os dados por um dos caminhos secundários, se existirem.

A disponibilidade dos caminhos é verificada através de pacotes chamados *HEARTBEAT*, enviados periodicamente em todos os caminhos secundários. Um congestionamento temporário ou falha permanente no caminho primário pode ocasionar a troca do fluxo de dados para um dos caminhos secundários. O método usado para troca de caminhos é chamada de *failover mechanism*. Outras aplicações do *multihoming*, como por exemplo o suporte a mobilidade na camada de transporte ou o balanceamento de carga através de múltiplos caminhos, não são explicitamente suportados na especificação padrão do SCTP.

#### 2.1.1.5 Multistreaming

*Multistreaming* é uma outra funcionalidade do SCTP que permite que as associações tenham múltiplas *streams*. *Stream* é um fluxo uni-direcional de dados em uma associação. O número de *streams* de uma associação é declarado na inicialização da comunicação e estas *streams* ficam ativas durante toda a conexão.

Cada *stream* é composta por um identificador que é incluído em cada *chunk* de dados, permitindo assim que *chunks* de várias *streams* sejam enviados em um mesmo pacote SCTP. São utilizados ainda números de sequência para cada uma das *streams*, assim na ocorrência de um erro, apenas a *stream* onde o erro foi detectado é afetada e não toda a associação. Estes números de sequência independentes reduzem a ocorrência de bloqueio (HOL) [23]. O bloqueio HOL pode ocorrer quando os protocolos de transporte oferecem um serviço de entrega confiável e ordenada pois, na ocorrência de uma perda, os segmentos subsequentes devem aguardar a confirmação de recebimento do segmento perdido, introduzindo atraso na entrega dos pacotes subsequentes.

#### 2.1.1.6 *Slow-Start*, controle de fluxo e de congestionamento

O controle de fluxo e de congestionamento do SCTP é essencialmente o mesmo do empregado no TCP. O *host* de origem mantém uma variável, chamada *advertized receiver window size* (*a\_rwnd*), para armazenar o valor restante de espaço no *buffer* de recebimento do *host* de destino. O *host* de destino mantém a variável *rwnd* (*receiver advertised window*) que mostra a quantidade de dados que ainda podem ser recebidos por toda a associação. O *host* de origem não pode enviar nenhum novo dado até que o *host* de destino indique que ele tem

espaço disponível para armazenar mais dados. Assim ambos os *hosts* conseguem controlar a quantidade de dados que são inseridos na rede.

O controle de congestionamento previne a sobrecarga da rede reduzindo a taxa de envio na ocorrência de perda de pacotes, que é considerada uma indicação de congestionamento da rede. Quando não é gerado nenhum evento de perda, o SCTP incrementa a sua taxa de envio. Se alguma perda for detectada, o SCTP divide a sua taxa de envio. Esta abordagem, chamada de *Additive Increase/Multiplicative Decrease* (AIMD), é composta de quatro principais estados: *slow start*, *congestion avoidance*, *fast retransmission* e *fast recovery*.

Ao iniciar a transmissão de dados ou após um longo período de inatividade, o SCTP não tem conhecimento sobre as condições da rede, requerendo o SCTP teste a rede a fim de determinar sua capacidade. O algoritmo *slow-start* é utilizado para este propósito no início de uma transmissão de dados ou após o re-estabelecimento de um caminho que apresentou perdas. Durante a execução deste algoritmo, a taxa de transmissão aumenta a cada pacote de confirmação recebido até que uma perda seja detectada, ou a janela de recebimento do *host* de destino (rwnd) esteja cheia, ou o valor do *Slow-Start threshold* seja alcançado. Uma vez alcançado o valor do *Slow-Start threshold*, o SCTP passa para o algoritmo de *congestion avoidance*.

O algoritmo de *congestion avoidance*, incrementa o valor da janela de controle de congestionamento (cwnd) por  $1 * \text{Maximum Transmission Unit (MTU)}$ , quando recebida a confirmação de entrega de um pacote enviado. Se ocorrer a perda de algum pacote, o SCTP assume este ocorreu por problemas de congestionamento na rede e diminui a taxa de envio de pacotes. A detecção de perda é realizada através da utilização de temporizadores, assim quando a confirmação de recebimento de um pacote não é recebida em um tempo específico, o *host* de origem assume que este segmento foi perdido e retransmite imediatamente o pacote que foi perdido, independente do valor de cwnd.

A variável *congestion control window* (cwnd) é mantida pelo *host* de origem, tendo por base cada um dos endereços de destino, e é ajustada conforme observações das condições da rede. Para distinguir entre as fases de *slow-start* e controle de congestionamento é utilizada a variável *sssthresh* (*slow-start threshold*), mantida pelo *host* de origem, tendo por base cada um dos endereços de destino.

Na ausência de perda de pacotes, o *host* de destino pode reconhecer múltiplos pacotes enviados anteriormente ao invés de reconhecer cada pacote individualmente. Entretanto, se o *host* de destino detecta uma falha em sua janela de recebimento, ele deve começar a enviar pacotes *Selective Acknowledgement* (SACK) para o *host* de origem todas as vezes que um pacote for recebido, até que os pacotes faltantes forem recebidos. Se o *host* de origem receber pacotes SACK indicando que existe algum pacote não reconhecido, ele deve aguardar por mais duas indicações de perda do mesmo pacote antes de retransmitir um pacote. Recebidas três indicações de não recebimento de algum pacote, o mesmo deve ser marcado para retransmissão e devem ser retransmitidos todos os pacotes não reconhecidos de um mesmo destino, em um novo segmento.

Se a transmissão não estiver executando o algoritmo de *fast recovery*, ele deve entrar neste modo e marcar o último pacote em sua janela de envio como seu ponto de saída. Quando for confirmado o recebimento de todos os pacotes, inclusive o do ponto de saída, a transmissão sai do modo *fast recovery*. Durante a execução do *fast recovery*, o valor de *Slow-Start threshold* e *congestion control window* é incrementado.

### 2.1.1.7 Mecanismo de troca de caminho

O mecanismo de detecção de falhas em uma associação SCTP *multihoming*, configurada para transferência confiável, age todas as vezes que ocorrer um estouro do temporizador RTO, incrementando o contador de *timeout*, duplicando o valor do RTO, reduzindo a janela de transmissão e retransmitindo o pacote. Este processo irá se repetir até que um ACK seja recebido ou até que o valor do contador de *timeout* seja maior que o valor de *Path Max Retransmissions* (PMR) cujo valor padrão é 5 [5]. Caso o contador de erros do caminho primário atingir o limite estabelecido na variável PMR, o caminho primário é considerado indisponível ou inalcançável e a troca de caminho é realizada, processo chamado *failover*. Por outro lado, se após a perda de um pacote, um SACK de outro pacote é recebido no caminho primário, o contador de erros é zerado e os *timeouts* anteriores considerados como causados por congestionamento temporário. O PMR influencia diretamente na velocidade de reação do SCTP para efetuar a troca caminhos na ausência de confirmação. Dessa forma, valores baixos de PMR correspondem a um comportamento mais agressivo do SCTP e valores altos configuram um comportamento

mais permissivo em relação a perdas.

Caso a sessão seja configurada para transmissão não confiável, independente do recebimento ou não de uma confirmação de recebimento, o *host* de origem irá prosseguir com a transmissão dos pacotes, ou seja, um novo pacote é transmitido após o recebimento de um ACK ou após o *timer* expirar.

Uma das primeiras avaliações sobre o mecanismo de troca de caminho (do inglês *failover*) do SCTP foi feita por Jungmaier, Rathgeb e Tüxen [24]. Examinando a performance do SCTP em aplicações SS7 verificou-se que os valores padrão do PMR, RTO Max e RTO Min do SCTP, relacionados com a detecção de falhas, levam a uma performance inaceitável para aplicações de sinalização. Isto ocorre porque os valores definidos por padrão no SCTP foram especificados para aplicações de uso geral e não de sinalização. Simulações examinando o efeito de diminuição do PMR bem como os valores de mínimo e máximo para o *timeout* de retransmissão (RTO.min e RTO.max), mostraram que os ajuste nestes parâmetros podem melhorar a performance de aplicações SS7.

Estudos foram feitos por Grinnemo e Brunstrom [25] a fim de verificar os efeitos do tráfego de fundo em rajadas e tamanho do *buffer* de roteadores em cenários de transporte de sinalização SS7 utilizando o SCTP. Os resultados mostraram que na presença de tráfego de fundo, grandes *buffers* em roteadores podem causar grandes variações de atraso, o que tende a aumentar o valor calculado de RTO no *host* de origem e também do tempo de *failover*.

Uma fórmula para estimar o tempo de *failover* no SCTP para cenários *multihoming* foi proposta por Budzisz et al. [26]. Os autores demonstraram que a estimativa normalmente utilizada na literatura, baseada apenas na soma de *timeouts* consecutivos, não é sempre apropriada, especialmente em cenários de mobilidade na camada de transporte.

Os efeitos positivos da redução do valor de PMR também foram mostrados nas simulações realizadas por Caro, Amer e Stewart [27] e Caro, Amer e Stewart [28]. Nestes trabalhos os autores mostraram melhora na performance da comunicação a medida que o valor de PMR era diminuído até zero, mesmo que o valor de PMR em zero ocasione trocas de caminho desnecessárias. Baseados nestes resultados os autores sugerem um mecanismo de troca de caminhos mais agressivo em que o tráfego de um caminho é imediatamente migrado para um novo caminho ativo na ocorrência de um *timeout*. O sistema então passa a checar a

conectividade no antigo caminho primário através de *heartbeats* e este não é marcado como inativo até que o PMR para aquele caminho seja excedido. Se houver sucesso no teste de conectividade, o tráfego volta para o caminho que antes era primário.

Conforme relatado por Noonan et al. [29] o mecanismo de *failover* do SCTP pode parar quando algumas condições não usuais ocorrerem, tal como uma estimativa errada de baixo valor de RTO para o caminho secundário ou erros de rede que afetem somente pacotes SACK. A solução mais geral proposta por Natarajan et al. [30] é de incluir um campo na SACK para identificar o caminho para o qual o reconhecimento de dados se refere, já que em caso de retransmissão, o SACK é enviado em ambos os caminhos.

### 2.1.1.8 PR-SCTP

O serviço de transporte parcialmente confiável (do inglês: *Partial Reliability*) (PR-SCTP), definido na RFC3758 [31], é uma extensão do SCTP que possibilita ao usuário especificar as regras sobre quão persistente o serviço de transporte deve ser, na tentativa de entregar uma mensagem ao destino. Em conjunto com o *multistreaming*, o PR-SCTP pode ser utilizado para promover um melhor suporte à aplicações de tempo real. Um exemplo de serviço parcialmente confiável é o chamado serviço com confiabilidade temporal que permite ao usuário indicar o limite de tempo para que a mensagem seja enviada ou retransmitida.

Para que o PR-SCTP funcione, foi adicionado um novo tipo de *chunk* no SCTP chamado FORWARD-TSN, que indica ao *host* de destino da mensagem que este deve avançar com o seu ponto de ACK cumulativo, possivelmente ignorando um ou mais pacotes de dados que até então não foram recebidos ou reconhecidos.

Em algum ponto da comunicação, o *host* de origem utilizando o PR-SCTP pode determinar que um pacote de dados em particular não deve ser enviado ou retransmitido. Neste caso este pacote deve ser marcado como abandonado e deve ser descartado e deve enviar uma mensagem de FORWARD-TSN para o *host* de destino para que este também marque o pacote como ignorado e desloque a sua janela de congestionamento.

Algumas das vantagens da utilização do PR-SCTP são:

1. Alguns protocolos de camada de aplicação podem se beneficiar por serem capazes de

utilizar uma única associação SCTP para transportar tanto dados que necessitam de garantia de entrega, tais como: páginas de texto, pagamentos e informações de conta bancária, sinalização de rede; e para serviços sem garantia de entrega como: dados que sejam sensíveis a atraso, onde a geração de um novo pacote é mais vantajosa que a retransmissão de um pacote antigo.

2. Os dados transportados pelo PR-SCTP terão as mesmas condições de detecção e proteção contra falhas que o serviço de transporte confiável do SCTP. Isto inclui a habilidade de detecção rápida de falha em um endereço de destino, troca de caminho para um endereço alternativo e ser notificado caso um destino se torne inalcançável.
3. O PR-SCTP pode prover uma combinação dos serviços de entrega confiável e ordenação de pacotes. Além de prover transporte de dados não ordenados e sem garantia de entrega, como feito pelo UDP, o PR-SCTP pode prover entrega confiável mas não ordenada.
4. O PR-SCTP emprega o mesmo controle de fluxo e de congestionamento em todo o tráfego, independente de este ser confiável ou parcialmente confiável.
5. Por conta da função de agrupamento de *chunks* do SCTP, dados de transmissões confiáveis e não confiáveis podem ser multiplexados em uma mesma associação SCTP. Assim, o número de datagramas IP pode ser reduzido, ao invés de se ter múltiplos tipos de dados sendo enviados por diferentes protocolos. Ainda, esta multiplexação permite que sejam utilizadas uma única porta, ao invés de se utilizar diferentes portas para o serviço confiável e não confiável.

#### **2.1.1.9 Balanceamento de carga**

Técnicas de balanceamento de carga permitem que os dados sejam divididos entre as diversas interfaces de rede. Elas têm um grande potencial de melhorar a performance da transmissão e eficiência da rede [32]. Apesar de todos os benefícios, um grande problema enfrentado a transferência simultânea de pacotes é a ordenação dos pacotes recebidos no *host* de destino.

O algoritmo de controle de congestionamento do SCTP, que foi derivado do TCP, pode apresentar problemas de performance em situações onde se necessita de constante ordenação dos pacotes. Assim o gerenciamento do *buffer* de envio e do controle de congestionamento

devem ser atualizados para levar em consideração os problemas de envio de dados através de múltiplos caminhos utilizando um único número de sequência e as consequências da reordenação de pacotes.

Uma das primeiras propostas para prover o balanceamento de carga no SCTP foi proposta por Ahmed, Saadawi e Lee [33], através de uma extensão para o SCTP chamado de *Load-Sharing SCTP* (LS-SCTP) [34]. O LS-SCTP separa o controle de fluxo, que é gerenciado por associação, do controle de congestionamento que, para fins de balanceamento de carga, deve ser gerenciado por caminho. Houve ainda a introdução de dois tipos de *chunks* adicionais para transportar dados e *acknowledgements* relacionados no LS-SCTP. Ambos *chunks* são compatíveis com os *chunks* do SCTP padrão, sendo a única diferença entre eles os números de sequência adicionados para facilitar o controle de congestionamento para os caminhos, que servem para rastrear os pacotes enviados a um dado destino. A solução proposta também oferece um mecanismo de monitoramento de caminho com *heartbeat* frequentes para evitar que a comunicação pare por conta de um caminho inativo. Apesar de o LS-SCTP poder parcialmente evitar os efeitos negativos causados pela reordenação de pacotes, a numeração adicional por caminho resulta em um consumo de banda desnecessário, pois informações similares podem ser inferidas através das variáveis de estado no *host* de origem e pelos SACK *chunks* no seu formato padrão.

Ao invés de utilizar explicitamente a numeração de pacotes por caminho, [35] propôs o *Independent per Path Congestion Control SCTP* (IPCC-SCTP) que provê um mapeamento local por caminho para cada pacote SCTP, uma vez que esta informação somente é necessária no *host* de origem para realizar o controle de congestionamento. Utilizando este mapeamento local, o SCTP pode realizar o controle de congestionamento, processamento de SACKs e retransmissão de pacotes por caminho, ao invés de realizar isto em toda a associação, como no SCTP padrão.

O IPCC-SCT foi seguido por outra abordagem de balanceamento de carga inicialmente proposto por Iyengar, Amer e Stewart [36] e descrito por completo em Iyengar, Amer e Stewart [37], chamada de *Concurrent Multipath Transfer* (CMT). Os autores propuseram que cada caminho tivesse *buffers* virtuais para garantir a independência entre eles. Tal abordagem garantiu a independência entre os caminhos mas teve implicações no controle de congestionamento. Para resolver este problema, algumas modificações no SCTP foram propostas:



- Foi proposto um algoritmo de crescimento da janela de congestionamento, realizando o controle de congestionamento por caminho e não por associação e limitando problemas de reordenamento de pacotes. SACKs recebidos são utilizados para atualizar o número de sequência do ponto de transmissão cumulativo (CumTSN), aumentando a janela de congestionamento por caminho, mesmo que estejam fora de ordem pela associação.
- Pequenas modificações no algoritmo de retransmissões rápidas, como o reordenamento introduzido pelo *host* de origem CMT, podem provocar falsas retransmissões rápidas com implicações na *cwnd*. A eliminação destas falsas retransmissões rápidas é realizada pelo algoritmo de *split fast retransmit* (SFR) que leva em consideração não somente a informação do SACK mas também o destino para cada número de sequência da transmissão (TSN) quando analisando a retransmissão para um dado caminho.
- O *host* de destino CMT não deve enviar imediatamente pacotes SACK para pacotes que chegaram fora de ordem, por conta de a rede ser vulnerável ao aumento de tráfego de ACK. Como o *host* de destino SCTP não distingue uma perda de uma reordenação, introduzido pelo *host* de origem CMT, um algoritmo chamado ACK atrasado para CMT é utilizado no *host* de origem para inferir perdas corretamente. No *host* de destino o algoritmo *dynamic IP address configuration* (DAC) estende o SACKs com informações sobre o número de pacotes de dados (PDUs) recebidos desde que o último SACK foi enviado.
- Cinco políticas de retransmissão de pacotes foram propostas por Iyengar, Amer e Stewart [38] mas, considerando aplicações que transmitem grandes quantidades de dados, os melhores resultados foram obtidos por políticas de retransmissão de pacotes para o caminho com maior tamanho de janela de congestionamento ou com maior valor de *Slow Start Threshold*.

O CMT é uma extensão do SCTP padrão que permite que o controle de congestionamento trabalhe com reordenação de pacotes. Entretanto a reordenação de pacotes ainda leva a um problema adicional que impacta o controle de fluxo. Este problema é conhecido como bloqueio de *buffer* no *host* de destino, onde o *buffer* é preenchido com dados fora de ordem por conta de falhas contínuas ou de curto prazo e, por conta de limitações no tamanho do *buffer* no *host* de destino, pode levar o algoritmo de controle de fluxo do *host* de origem a não realizar novas

transmissões de dados.

O bloqueio de *buffer* no *host* de destino foi estudado nos trabalhos de Iyengar, Amer e Stewart [39] e Iyengar, Amer e Stewart [40] onde foram sugeridas novas políticas para controle de congestionamento. Uma solução que mitiga parcialmente o problema de bloqueio de *buffer* no *host* de destino, chamada CMT *Potentially Failed* (CMT-PF), foi apresentada por Natarajan et al. [30] e posteriormente normatizada pela RFC7829 [41]. O CMT-PF marca o caminho onde foi encontrada a falha como potencialmente falho e para a transmissão de dados naquele caminho até que um retorno positivo de um pacote *Heartbeat* seja recebido. Uma extensa avaliação do CMT-PF foi apresentada por Preethi et al. [42], onde o autor mostra que a solução proposta apresenta melhor ou similar, mas não inferior, ao CMT padrão.

#### 2.1.1.10 *delay-centric*

O monitoramento do atraso no SCTP é feito através da aferição do RTT nos diferentes caminhos, através dos pacotes de *Heartbeat* nos caminhos alternativos ou do próprio tráfego no caso do caminho primário. Esta informação é utilizada como base para atualizar as variáveis SRTT, RTTVAR e RTO, conforme apresentado na seção 2.1.1.2, que têm papel importante no algoritmo de *failover* e controle de congestionamento.

Originalmente proposto por Kelly et al. [14], o *delay-centric* foi desenvolvido para aumentar a qualidade no transporte de dados sensíveis ao atraso na internet. Ele funciona em conjunto com o SCTP e se baseia em dados sobre o atraso apresentado por cada um dos caminhos envolvidos na associação. Baseado nestes dados, o *delay-centric* altera o caminho primário automaticamente para o caminho que apresente o menor atraso.

Conforme mostrando em [18], podem ocorrer situações onde a troca constante de caminho leve a um efeito manada, onde todos os agentes trocam de caminho simultaneamente, levando a um problema de bloqueio de *buffer* no *host* de destino assim como observado no CMT. Para mitigar esse problema, o *delay-centric* pode utilizar um parâmetro de histerese, com o objetivo de não permitir a troca de caminho enquanto o menor valor de SRTT mais o valor de histerese, seja maior que o valor de SRTT do caminho primário.

## 2.2 Diversidade de caminhos

A convergência de redes de dados e voz, e o crescimento de dispositivos de rede inteligente são vistos como duas principais tendências que irão acelerar a introdução de um maior número de aplicações e serviços na internet. Em redes de Voz sobre IP ou vídeo conferência, mesmo pequenas falhas ou atrasos podem degradar a qualidade de serviços. Mesmo sendo crítico garantir um alto nível de disponibilidade fim-a-fim, os atuais protocolos de roteamento são conhecidos por serem lentos para reagir e se recuperar de falhas na rede, fazendo com que estas falhas sejam visíveis aos *hosts* finais.

Conforme Rohrer [6], a diversidade é uma característica essencial de topologias de rede que permite melhorar a resiliência em comunicações fim-a-fim, consequentemente aumentando níveis de qualidade de serviço, em face de eventos diversos.

A diversidade de caminhos é diretamente influenciada pela diversidade física e lógica na forma de nós de roteamento e *links* entre destinos finais. A diversificação de caminhos tem como alvo a comunicação fim-a-fim, mas pode ser aplicada em qualquer camada de rede, onde o serviço de escolha de caminhos esteja disponível, podendo utilizar como métrica de performance, requisitos como baixos níveis de latência ou disponibilidade.

Os benefícios da utilização de múltiplos caminhos podem ser observados na forma de melhora da performance, aumento da disponibilidade ou ambos. O objetivo da diversificação de caminhos é prover uma interface única para um serviço que tenha a mesma disponibilidade quanto o grafo da rede física, ao invés de estar limitado pela confiabilidade de um único caminho como no atual modelo. O serviço fim-a-fim deve ser informado sobre os requisitos da aplicação em termos de *throughput* e máximo atraso permitido, sabendo que existe uma relação inversa entre maximizar a diversidade e minimizar a distância destes caminhos.

Para mensurar a diversidade de caminhos e aumentar a resiliência, a escolha dos caminhos deve ser de forma tal que eles não experimentem falhas correlacionadas. Para este fim, a medida de diversidade, originalmente introduzida em Rohrer, Naidu e Sterbenz [43] e redefinido em Rohrer, Jabbar e Sterbenz [44], é utilizada para quantificar o grau em que os caminhos compartilham nós e *links*.

A diversidade é medida conforme as seguintes definições:

- **Caminho:** Definido como o par de nós (origem s, destino d), o caminho P entre eles é um vetor contendo todos os *links* L e todos os nós N intermediários deste caminho.
- **Diversidade de caminho:** Seja o caminho mais curto entre um dado par (s,d) denominado  $P_0$ . Então, para qualquer caminho alternativo  $P_k$  entre a origem e o destino, a função de diversidade  $D(x)$  em relação a  $P_0$  é definida como:

$$D(P_k) = 1 - \frac{|P_k \cap P_0|}{|P_0|} \quad (2.8)$$

A diversidade de caminho é 1 se  $P_k$  e  $P_0$  forem completamente disjuntos e o valor 0 quando  $P_k$  e  $P_0$  são idênticos.

Para dois caminhos arbitrários  $P_a$  e  $P_b$  a diversidade de caminhos é definida como:

$$D(P_b, P_a) = 1 - \frac{|P_b \cap P_a|}{|P_a|} \quad (2.9)$$

Onde  $|P_a| \leq |P_b|$

Essa metodologia pode ser utilizada para analisar vários tamanhos de rede. Por exemplo, no contexto de uma rede WAN (*Wide Area Network*) que possui eventos e conexões em larga escala geográfica, um PoP (*Point of Presence*) pode ser representado como um nó e um *link* pode ser representar um conjunto de fibras em um mesmo cabo óptico que o interliga a outro PoP.

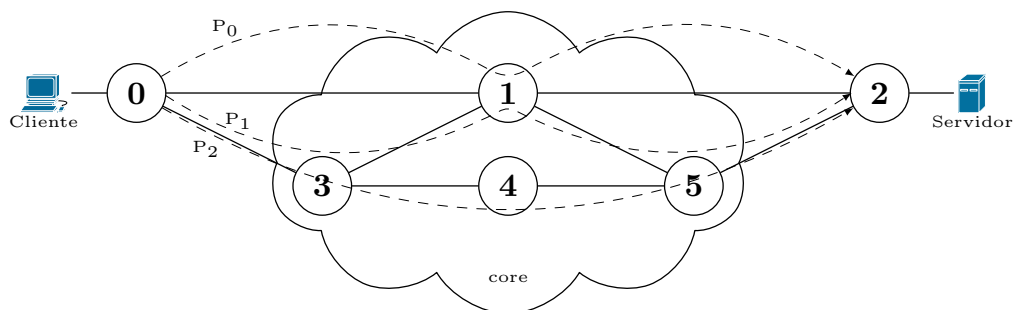


Figura 2.1: Diversidade de Caminhos

Como exemplo, consideremos o cenário da figura:2.1. Este cenário é composto por um

cliente e um servidor. Os nós 0 e 2 representam as redes de acesso do cliente e servidor respectivamente. Os nós 1,3,4 e 5 compõe o core da rede. Entre as redes de acesso do cliente e do servidor existem três caminhos:  $P_0$ ,  $P_1$  e  $P_2$ , identificados pelas linhas tracejadas. As linhas sólidas identificam ligações lógicas entre os nós, podendo conter mais de um meio de acesso físico. Para representar os nós, foi utilizada a nomenclatura  $N_i$  onde  $i$  representa o nó em questão. Para representar os links, foi utilizada a nomenclatura  $L_{i,j}$ , onde  $i$  representa o nó de origem e  $j$  o nó de destino. Desta forma podemos identificar que:

- O caminho  $P_0$  é composto pelos nós: 0, 1 e 2 e pelos links  $L_{0,1}$  e  $L_{1,2}$ .
- O caminho  $P_1$  composto pelos nós: 0,3,1,5,2 e pelos links  $L_{0,3}$ ,  $L_{3,1}$ ,  $L_{1,5}$  e  $L_{5,2}$ .
- O caminho  $P_2$  composto pelos nós: 0,3,4,5,2 e pelos links  $L_{0,3}$ ,  $L_{3,4}$ ,  $L_{4,5}$  e  $L_{5,2}$ .

Pela equação 2.8, podemos calcular o  $D(P_1)$  como:

$$D(P_1) = 1 - \frac{|\{N_3; N_1; N_5; L_{0,3}; L_{3,1}; L_{1,5}; L_{5,2}\} \cap \{N_1; L_{0,1}; L_{1,2}\}|}{|\{N_1; L_{0,1}; L_{1,2}\}|} \quad (2.10)$$

$$D(P_1) = 1 - \frac{|\{N_1\}|}{|\{N_1; L_{0,1}; L_{1,2}\}|} \quad (2.11)$$

Assumindo cada nó e cada Link com valor 1:

$$D(P_1) = 1 - \frac{1}{3} = \frac{2}{3} \approx 0.66667 \quad (2.12)$$

Da mesma forma, observa-se que o valor de  $D(P_2) = 1$ , pois ele não apresenta nenhum nó ou link em comum com o  $P_0$ .

Utilizando a equação 2.9, podemos calcular o valor de  $D(P_1, P_2)$  como:

$$D(P_1, P_2) = 1 - \frac{|\{N_3; N_1; N_5; L_{0,3}; L_{3,1}; L_{1,5}; L_{5,2}\} \cap \{N_3; N_4; N_5; L_{0,3}; L_{3,4}; L_{4,5}; L_{5,2}\}|}{|\{N_3; N_4; N_5; L_{0,3}; L_{3,4}; L_{4,5}; L_{5,2}\}|} \quad (2.13)$$

$$D(P_1, P_2) = 1 - \frac{|\{N_3; N_5; L_{0,3}; L_{5,2}\}|}{|\{N_3; N_4; N_5; L_{0,3}; L_{3,4}; L_{4,5}; L_{5,2}\}|} \quad (2.14)$$

$$D(P_1, P_2) = 1 - \frac{4}{7} = \frac{3}{7} \approx 0.42857 \quad (2.15)$$

Com estes cálculos, conclui-se que o melhor caso é a utilização de  $P_0$  e  $P_2$ , por serem completamente disjuntos, ou terem uma diversidade de 100%, seguido pela utilização de  $P_0$  e  $P_1$  com diversidade de 66,67% e o pior cenário é a utilização de  $P_1$  e  $P_2$  com diversidade de 42,87%.

### 2.2.1 Diversidade de caminhos da internet

Segundo Han, Watson e Jahanian [13] para oferecer melhor performance e disponibilidade, um número de abordagens foram propostas para utilizar de maneira efetiva a redundância da internet. A ideia de que sistemas baseados em *multihoming* e redes sobrepostas podem prover significativos ganhos de disponibilidade, somente é válida quando os caminhos atravessando diferentes operadoras possuem alto nível de diversidade. Neste sentido, os autores analisando de maneira quantitativa o impacto da diversidade de caminhos, demonstraram limitações do controle de rotas do *multihoming* e arquiteturas sobrepostas sobre várias perspectivas, quantificando a extensão da diversidade de caminhos e identificaram a origem de limitações em redes *multihoming*, além de examinar o impacto da diversidade de caminhos em redes sobrepostas e suas limitações.

Na figura 2.2, o eixo vertical se refere à distribuição cumulativa de destinos e o eixo horizontal se refere ao número de roteadores compartilhados. Esta figura mostra que nos testes realizados por Han, Watson e Jahanian [13], mostra que a quantidade de caminhos que não possuem nenhum roteador em comum em *hosts* que possuem duas alternativas de conexão com a internet é de aproximadamente 20%. Em *hosts* que possuem de três a dez conexões com a internet a quantidade de caminhos que não possuem nenhum roteador em comum fica entre 40% e 50%. Considerando que exista algum nível de compartilhamento de nós, percebe-se que não há diferença aparente na distribuição cumulativa, quando se possui mais do que quatro caminhos alternativos com a internet.

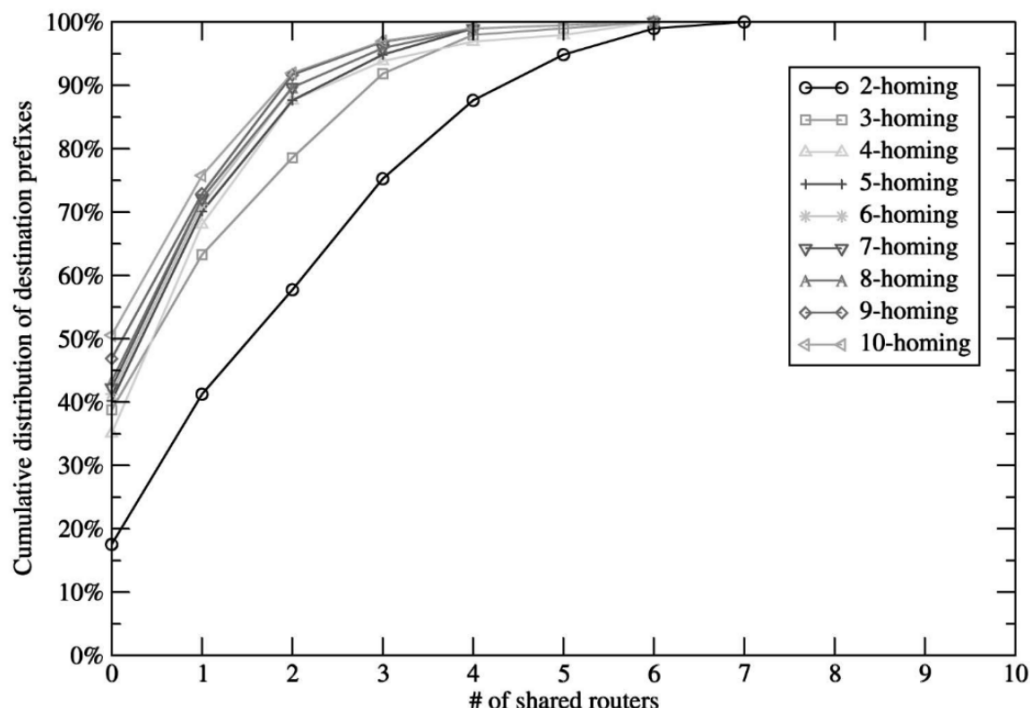


Figura 2.2: Diversidade de caminhos na Internet. Fonte Han, Watson e Jahanian [13]

Ainda sobre a figura 2.2 verifica-se que em *hosts* que possuem de três conexões com a internet, 60% destes ou são completamente disjuntos ou possuem pelo menos um nó de roteamento em comum. Isto mostra que para se ter uma melhor aproximação com a realidade, simulações que considerem transmissão de dados pela internet, devem possuir cenários com caminhos não completamente disjuntos, ou seja, considerando a existência de pelo menos um nó de roteamento em comum.

### 2.3 Medição da qualidade da internet

Nos Estados Unidos o *Federal Communications Commission* (FCC) elabora um documento anual intitulado *Measuring Broadband America* onde são apresentados dados sobre a qualidade da internet, separadas por tecnologia de acesso e por operadoras. Até o último relatório, do ano de 2016 [45], somente eram analisados os dados referentes a serviços de internet fixa. O mesmo formato de documento é elaborado pela *European Commission* (EC), intitulado *Quality of Broadband Services in the EU* [46].

Consultando os dados sobre velocidade média de recebimento de dados pelos usuários

por tecnologia na Tabela 2.1, verifica-se que a velocidade média, independente da tecnologia é de 41.23 Mbps. Se analisada a média de velocidade por tecnologia, observa-se que o DSL (*Digital Subscriber Line*) possui média de 6.76 Mbps, cabo 49.16 Mbps, fibra óptica 38.21 Mbps e satélite 10.37 Mbps.

Tabela 2.1: Velocidade de *Download* por operadora Fonte: Federal Communications Commission (FCC) [45]

		<b>Setembro 2015</b>
<b>DSL</b>	AT&T - DSL	3.81
	AT&T - IPBB	15.38
	CenturyLink	10.07
	Frontier DSL	3.54
	Verizon DSL	2.48
	Windstream	5.29
<b>Cabo</b>	Optimum	44.84
	Charter	69.24
	Comcast	60.73
	Cox	46.13
	Mediacom	35.36
	TWC	38.63
<b>Fibra optica</b>	Frontier Fiber	21.95
	Verizon Fiber	54.47
<b>Satélite</b>	Hughes	12.19
	Viasat/Exede	8.56
<b>Todas as tecnologias</b>	Média de todas as operadoras	41.2302979294404

O relatório da EC [46] por sua vez mostra que em toda a União Europeia (EU), 79% dos usuários possuem mais de 10Mbps de velocidade de acesso à internet, independente da tecnologia. A tabela 2.2 entretanto mostra que 66% dos usuários possuem uma velocidade de acesso à internet de até 30Mbps, não especificando a tecnologia de acesso. Na tabela 2.3 é apresentada a quantidade de usuários por tecnologia na União Europeia, onde observa-se a predominância da utilização do DSL em detrimento de outras tecnologias.



Tabela 2.2: Porcentagem de usuários que possuem determinada taxa de transmissão na União Europeia. Fonte: [46]

	Até 30Mbps	Entre 30Mbps e 100Mbps	Acima de 100Mbps
<b>EU</b>	66%	20%	14%

Tabela 2.3: Porcentagem de usuários por tecnologia de acesso na União Européia. Fonte: European Commission (EC) [46]

	DSL	Cabo	Fibra optica	Outro
<b>EU</b>	68%	19%	10%	3%

No Brasil, as prestadoras de serviço oferecem o acesso à internet nas tecnologias DSL e cabo, sendo que a disponibilidade da tecnologia pela operadora e a velocidade contratada determinam qual das tecnologias será instalada para o usuário. Nos Estados Unidos as operadoras em sua maioria fornecem a seus usuários o acesso via cabo. Na União Europeia a maioria dos usuários possuem acesso via DSL. Ambas tecnologias entretanto possuem taxas de transferência e qualidade de serviço muito próximas.

## 2.4 NS2: *Network Simulator 2*

O *Network Simulator 2* (NS2) é um simulador de redes frequentemente utilizado em pesquisas devido à sua extensibilidade, pois é um software de código aberto escrito em C++. Nele já estão implementados diversos elementos e protocolos, permitindo assim a simulação de uma grande variedade de situações. Além disso ele apresenta geradores de tráfego, serviços e outras ferramentas que permitem a elaboração de cenários complexos sem a necessidade de novas implementações no código fonte.

Os *scripts* contendo a descrição da rede, como: topologia, agentes, serviços e os parâmetros a serem simulados, devem ser escritos na linguagem Otcl. O NS2 interpreta este *script*, executa a simulação e imprime em tela ou gera arquivos com os resultados.

## 2.5 Compressão de vídeo MPEG

O MPEG-4 parte 10 ou H.264 [47] é um padrão de codificação de vídeo, resultado de um projeto de cooperação entre a *Video Coding Experts Group* (VCEG) da *International Telecommunication Union* (ITU) e o *Moving Picture Experts Group* (MPEG).

Um vídeo de alta definição (1080p) compactado com H.264 com parâmetros padrões de compactação, necessita em média de uma taxa de transmissão de 8Mbps [48]. O H.264 explora redundâncias espaciais e temporais para compactar o vídeo. A redundância espacial geralmente denominada compactação *intraquadros*, define-se como semelhanças em um mesmo quadro. A compactação espacial é realizada através da transformada do cosseno *Discrete Cosine Transform* (DCT) [49]. A redundância temporal define-se como semelhança entre quadros. A compactação temporal ou também denominada compactação *interquadros* identifica as diferenças entre os quadros de um *Group of Pictures* (GOP) e possibilita que os quadros sejam descritos com base nas diferenças dos quadros anteriores.

Segundo Greengrass, Evans e Begen [50], para explorar a redundância temporal, o MPEG utiliza uma sequência pré definida de GOP, sendo o GOP formado por três tipos de quadros distintos. O quadro I (*intra*) é codificado sem a referência de outros quadros e usa apenas compressão espacial. O quadro P (*predictive*) é codificado a partir de um quadro I ou quadro P anterior. O quadro B (*bi-directionally*) utiliza informações do quadro P ou I anterior e do próximo quadro I ou P como referência para sua codificação.

O GOP tem início com um quadro I, que é seguido de diversos quadros B e P, como ilustrado na Figura 2.3. Para representação da estrutura de composição de quadros, é comum utilizar a notação  $(M, N)$ , onde M representa a quantidade de quadros por GOP e N representa a quantidade de quadros B consecutivos. Por exemplo  $(12, 2)$ , denota uma estrutura para um grupo de figuras formado pela sequência de quadros *IBBPBBPBBPBB*, que será repetida continuamente ao longo da duração do vídeo.

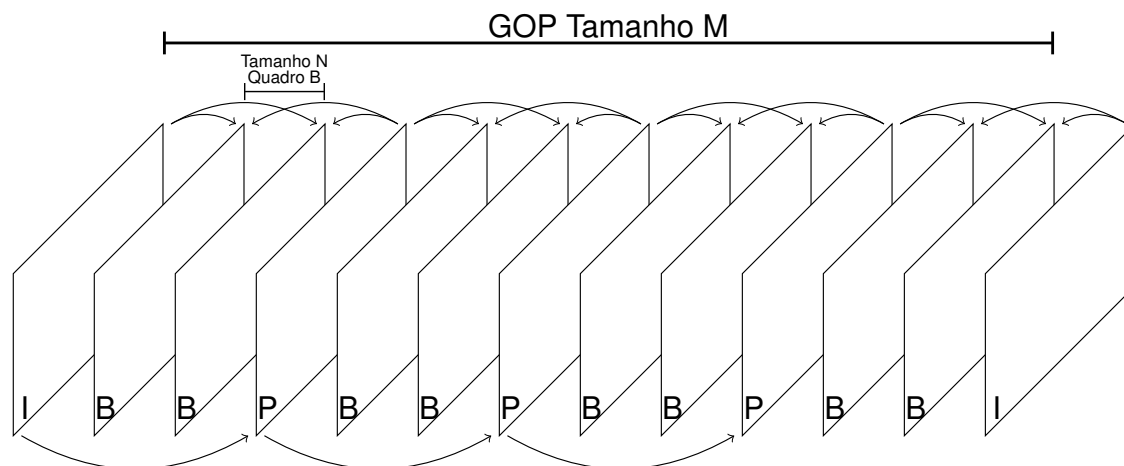


Figura 2.3: Sequência de formação do GOP

Neste capítulo inicialmente foi apresentado alguns conceitos fundamentais do protocolo SCTP e suas extensões, tais como o PR-SCTP, o CMT e o *delay-centric*. Foram ainda apresentados conceitos sobre diversidade de caminhos e as velocidades típicas de usuários dos Estados Unidos, União Europeia e Brasil. Em seguida foi descrito o funcionamento do simulador de redes NS2 e por último apresentados conceitos sobre compactação de vídeo em H.264. No próximo capítulo será apresentada a metodologia utilizada neste trabalho, contendo os desenvolvimentos realizados no NS2, cenários de simulação utilizados, métricas de desempenho e como foi realizada as coletas dos dados das simulações.

## CAPÍTULO 3

### ANÁLISE DE DESEMPENHO DE SISTEMAS MULTI-CAMINHO

Para avaliar a eficiência de sistemas multi-caminho em situações com competição de fluxos de vídeo H.264 em diferentes níveis de diversidade de caminhos e a influência da estabilidade do *delay-centric* nessas transmissões, foram propostos alguns cenários a fim de realizar a simulação com diversos níveis de diversidade, utilizando vídeo de alta resolução com diferentes características de compactação. Foi utilizado o filme Aspen<sup>1</sup>, compactado no formato H.264 com GOP (60,3) e com GOP (15,2), a 30 quadros por segundo perfazendo um total de 19 segundos.

As figuras 3.1 mostram o *throughput* médio dos vídeos Aspen com GOP(60,3) e Aspen com GOP(15,2) respectivamente, em relação ao tempo de transmissão. O *throughput* necessário para transmissão de cada vídeo, foi obtido através do software MP4trace utilizando o protocolo UDP. A diferença no *throughput* entre os vídeos se dá por conta da maior quantidade de quadros I, P e B, conforme mostra a tabela 3.1.

Tabela 3.1: Tabela comparativa de característica de GOP

GOP	Quadros I	Quadros P	Quadros B
(60,3)	10	116	336
(15,2)	38	190	342

<sup>1</sup>Video elaborado pelo *National Telecommunications & Information Administration/Institute for Telecommunication Sciences* (NTIA/ITS) no projeto 3141012-300 disponível em <http://media.xiph.org/video/derf>

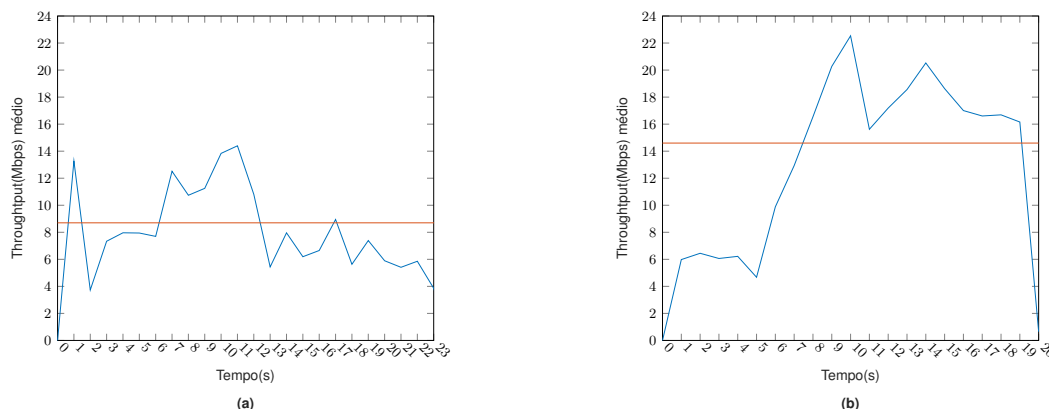


Figura 3.1: *Throughput* vídeo Aspen vs. tempo sendo a figura (a) GOP(60,3) e a figura (b) GOP(15,2)

A fim de tornar os cenários simulados mais próximos de um cenário real, todas as simulações as transmissões de vídeo iniciaram em um intervalo de tempo aleatório, seguindo distribuição uniforme com valores entre 0s e 10s. Isso foi feito no arquivo utilizado pelo NS2 para configurar a simulação utilizando o código constante no anexo 1.

Para se evitar a correlação entre os vários fluxos de vídeo, passou-se a, além de se iniciar a transmissão do vídeo em um tempo aleatório entre 0 e 10s, iniciar a transmissão a partir de um ponto aleatório uniformemente distribuído entre o primeiro e o último pacote do vídeo. Essa estratégia tem por objetivo dificultar a ocorrência de correlação entre os fluxos, conforme pode ser observado na figura 3.2.

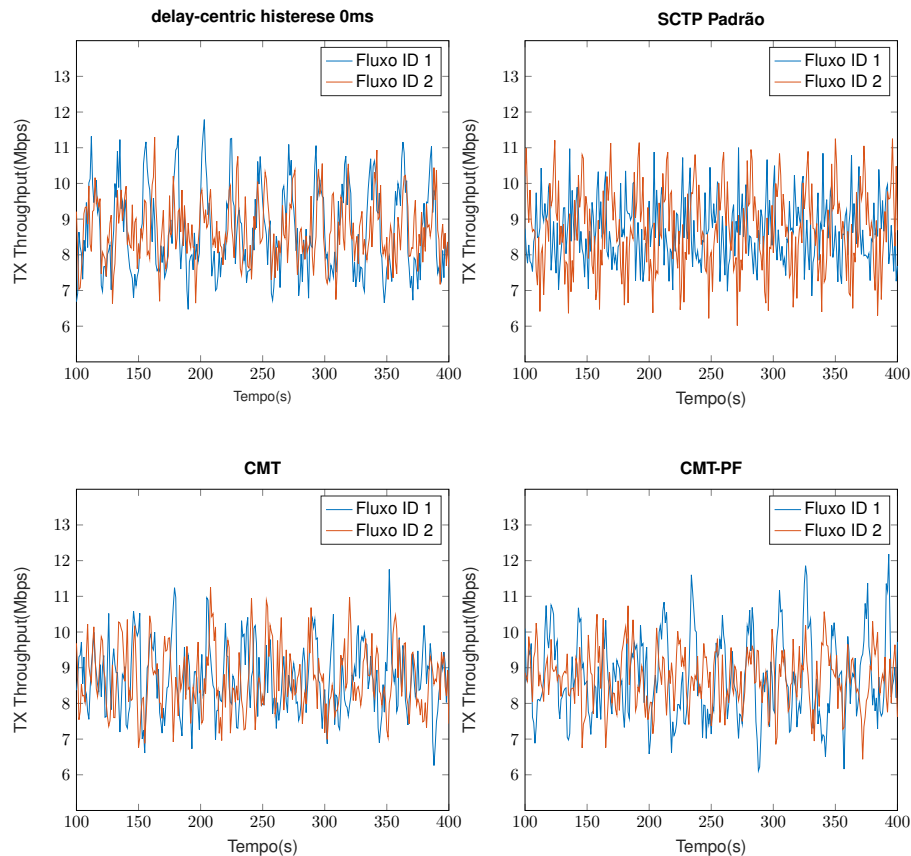


Figura 3.2: Gráfico do *Throughput* TX médio vs. tempo de simulação, dos fluxo ID 1 e 2, com início aleatório entre o primeiro e o último pacote do vídeo.

O fato de se iniciar o vídeo em instantes e posições diferentes possibilita a utilização de um único trace de vídeo para as simulações, pois torna cada um dos fluxos de vídeo únicos, como se fossem diferentes vídeos sendo transmitidos ao mesmo tempo.

### 3.1 Implementação no NS2

Por causa do *multihoming* alguns dos recursos do SCTP não são suportados por muitas aplicações de geração de tráfego disponíveis para o TCP e UDP. Por isto o usuário do simulador deve implementar o gerador de tráfego utilizando a *Application program interface* (API). Para todas as simulações foram utilizadas as configurações apresentadas na Tabela 3.2. As configurações especificam entrega não confiável, sem ordenação e sem controle de congestionamento.

Tabela 3.2: Parâmetros Utilizados na API de geração de tráfego

Parâmetro	Descrição	Padrão	Utilizado
numUnreliable	Número de <i>streams</i> não confiáveis	0	1
numStreams	Número total de <i>streams</i>	1	1
reliability	Confiabilidade de entrega da mensagem	1	0
unordered	Entrega de mensagens fora de ordem	0	1

Utilizando conceitos de herança e polimorfismo da Orientação a Objetos, o *delay-centric* foi implementado no NS2 diretamente no código C/C++. A função *SendMuch()* é a responsável pelo enfileiramento de pacotes nas interfaces, sendo acionada pelas funções *sendmsg()* ou *recv()*. A função *sendmsg()* é a responsável pelo recebimento de pacotes vindos da camada de aplicação. A função *recv()* é a responsável pelo recebimento de pacotes vindos da camada de rede.

Utilizando o conceito de polimorfismo, a função *SendMuch()* foi reimplementada, possibilitando que ao ser chamada, seja percorrida a lista de destinos para encontrar o que possua o menor valor de *srtt*. Se o caminho encontrado for diferente do primário e o valor de seu *srtt* seja maior que zero, realiza-se a troca de caminho e é chamada a função *sendmsg()* para que os dados sejam então enviados para a camada de rede do caminho primário, para serem transmitidos. A figura 3.3 mostra a implementação deste algoritmo.

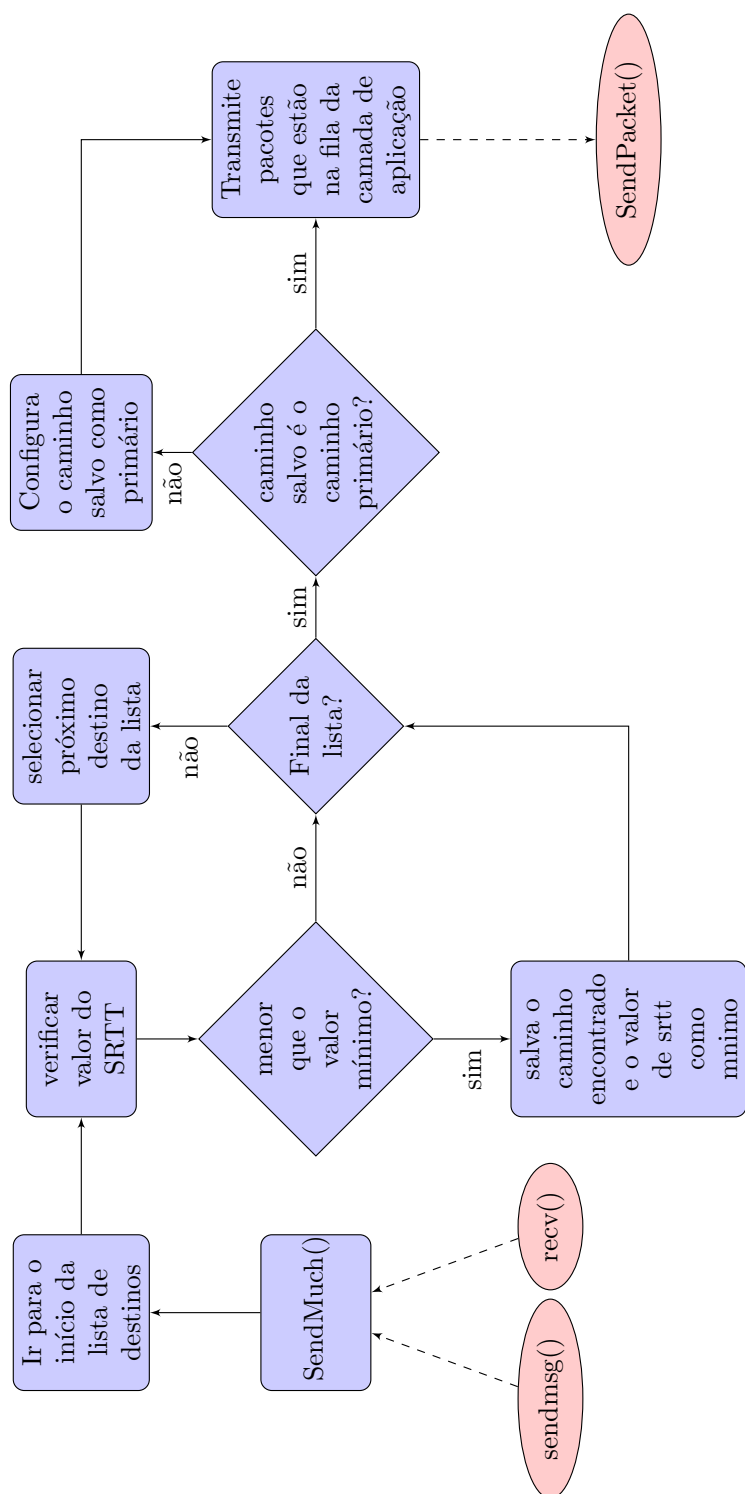


Figura 3.3: Fluxograma de implementação do delay-centric no NS2



## 3.2 Métricas de desempenho

Para avaliação de desempenho foram utilizadas métricas como atraso, vazão de transmissão (TX) e recepção (RX), *goodput*, índice de justiça na divisão do *throughput*. Para o *delay-centric* foi ainda mensurado a quantidade média de trocas de caminho realizadas para cada nível de ocupação.

### 3.2.1 Coleta dos dados

Durante a simulação o NS2 grava os eventos gerados por cada um dos nós para um arquivo. A tabela 3.4, baseada em [51], ilustra o formato deste arquivo de saída. Os tipos de evento são mostrados na tabela 3.3.

Tabela 3.3: Tipos de eventos

Evento	Descrição
r	pacote recebido pelo próximo nó
+	pacote enfileirado
-	pacote retirado da fila
d	pacote descartado

Com os eventos gerados pelo NS2, é possível adquirir as métricas de atraso, *throughput* e perda de pacotes de cada um dos fluxos, identificados pelo campo FID (Identificador do fluxo). Todos os nós criados na simulação geram os eventos mostrados na tabela 3.3 para cada pacote que seja recebido e/ou transmitido.

Tabela 3.4: Exemplo de formato do arquivo de *trace*

Evento	tempo	Origem do evento	Destino do evento	tipo de pacote	tamanho	opções	fid	Nó de origem . porta	Nó de destino . porta	Número de sequencia	id do pacote
r	1.3556	3	2	ack	40	---	1	3.0	0.0	15	201
+	1.3556	2	0	ack	40	---	1	3.0	0.0	15	201
-	1.3556	2	0	ack	40	---	1	3.0	0.0	15	201
r	1.35576	0	2	tcp	1000	---	1	0.0	3.0	29	199
+	1.35576	2	3	tcp	1000	---	1	0.0	3.0	29	199
d	1.35576	2	3	tcp	1000	---	1	0.0	3.0	29	199
+	1.356	1	2	cbr	1000	---	2	1.0	3.1	157	207
-	1.356	1	2	cbr	1000	---	2	1.0	3.1	157	207

As métricas de performance coletadas diretamente do arquivo de trace, para cada um dos fluxos, são armazenadas em vetores. Estas métricas de performance foram coletadas da seguinte forma:

As métricas de performance coletadas diretamente do arquivo de trace foram:

1. Atraso fim-a-fim: Métrica é obtida utilizando dois eventos, quando o pacote é recebido pelo nó de origem e quando este é retirado da fila do nó de destino para ser entregue à aplicação. Subtraídos estes tempos, têm-se o atraso individual para cada um dos pacotes.
2. *Throughput* TX: É obtida pela soma do tamanho de todos os pacotes retirados da fila do nó de origem em um determinado intervalo de tempo (foi usado 1 segundo).
3. *Throughput* RX: É obtida pela soma do tamanho de todos os pacotes recebidos pelo nó de destino em um determinado intervalo de tempo (foi usado 1 segundo).
4. A perda de pacotes é detectada quando um pacote é enviado pelo nó de origem, mas não é detectada a chegada deste pacote no destino. Pelas características da simulação, não foram detectadas perdas pois o tamanho dos *buffers* de entrada e saída dos nós de roteamento utilizados foram muito grandes, assim como são atualmente nos nós de roteamento da internet [13] [43] [44].

A partir dos dados coletados e armazenados em vetores, destas métricas coletadas diretamente do arquivo de trace, é possível calcular os valores médios por unidade de tempo (foi utilizado 1 segundo) do atraso e do *throughput*. A partir destes valores médios, calcula-se o erro médio quadrático, tanto do atraso quanto do *Throughput* médio, o índice de justiça na divisão do *throughput*, perda de pacotes e *goodput*.

### 3.2.2 Variância do atraso

O valor do atraso e *Throughput* médio é armazenado em um vetor. Estes dados são utilizados posteriormente para o cálculo da variância do atraso, através da equação 3.1 para cada intervalo de tempo:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\bar{d} - d_i)^2 \quad (3.1)$$

Onde  $i$  é uma variável utilizada na função contadora representando um segundo de simulação,  $n$  é o número de elementos armazenados no vetor,  $\bar{d}$  é o valor médio dos valores armazenados no vetor e  $d_i$  é o valor do atraso médio no segundo  $i$ .

### 3.2.3 Índice de Justiça de Jain

Em situações de competição por múltiplos agentes por um mesmo recurso é fundamental que este recurso seja compartilhado de maneira justa entre todos os agentes. O índice de justiça de Jain, definido pela equação 3.2, é utilizado para mensurar esta justiça entre vários agentes. A medida de justiça é feita considerando um conjunto de valores onde existem  $n$  fluxos e  $v_i$  é o *throughput* médio para o fluxo  $f_n$ . O resultado varia entre  $\frac{1}{f_n}$  (pior caso) e 1 (melhor caso), sendo que o melhor caso mostra que todos os fluxos compartilham de maneira igualitária os recursos.

$$\mathfrak{J}(v_1, v_2, \dots, v_n) = \frac{(\sum_{i=1}^{f_n} v_i)^2}{n \cdot (\sum_{i=1}^{f_n} v_i^2)} \quad (3.2)$$

### 3.2.4 Goodput

O *goodput* é calculado através da soma do tamanho dos pacotes que possuam atraso médio inferior a um limite, foi usado 200 milissegundos. Conforme descrito por Shen, Schulzrinne e Nahum [52], a não confirmação de recebimento de um pacote por mais de 200ms levam a situações de retransmissões desnecessárias, pois a aplicação no destino já pode ter marcado este pacote como perdido e seu recebimento não se faz mais necessário. A retransmissão de um pacote desnecessariamente em situações de congestionamento de rede somente agravam o problema.

### 3.3 Cenários utilizados nas simulações

A Figura 3.4 mostra a topologia de rede base que foi utilizada. Esta topologia é composta por nós servidores e nós clientes, redes de acesso que se interligam com clientes e servidores e o núcleo da rede que interliga as diferentes redes de acesso. A origem do fluxo de dados se dá nos nós servidores.

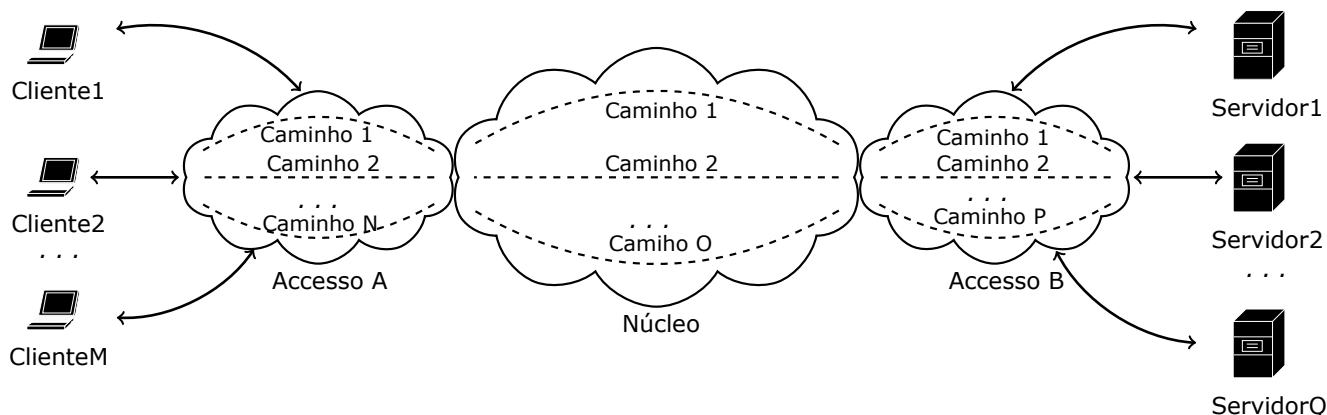


Figura 3.4: Cenários de simulação

A topologia mostrada na figura 3.4, adaptada da norma *G.1050 : Network model for evaluating multimedia transmission performance over Internet Protocol* ITU-T [53], visa reproduzir situações típicas de configurações de redes de acesso e núcleo em uso atualmente.

Os cenários propostos para simulação foram elaborados entendendo que haveria a possibilidade de atendimento ao cliente por duas ou mais redes de acesso na mesma localidade. Utilizando identificadores numéricos para cada nó de acesso, nós pares (incluindo os com índice zero) proveriam interfaces de 25Mbps e nós ímpares proveriam acessos de 50Mbps.

Cenário 1: Dois caminhos completamente disjuntos no núcleo da rede.

Cenário 2: Três caminhos completamente disjuntos no núcleo da rede.

Cenário 3: Três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core.

### 3.3.1 Cenário 1: Dois caminhos completamente disjuntos

A figura 3.5 apresenta o modelo de cenário de simulação, que é constituído de dois caminhos disjuntos no núcleo da rede. As linhas pontilhadas em vermelho e azul identificam estes caminhos. As interfaces dos clientes e servidores estão ligadas de maneira alternada aos nós de acesso que por sua vez, estão conectados aos nós do núcleo da rede.

O caminho identificado pela cor vermelha é composto pelos nós  $Ac_0$ ,  $Cr_0$  e  $As_0$ . Ao nó de acesso  $Ac_0$  estão ligadas a interface 0 do cliente 0 e a interface 1 do cliente 1. Ao caminho identificado pela cor azul pertencem os nós  $Ac_1$ ,  $Cr_1$  e  $As_1$  e a este caminho estão conectadas a interface 1 do cliente 0 e a interface 0 do cliente 1.

A forma como as interfaces foram conectadas aos nós de acesso permite que os agentes realizem a transmissão do vídeo por caminhos distintos, mas isto irá depender das condições de rede e do método de seleção do caminho primário.

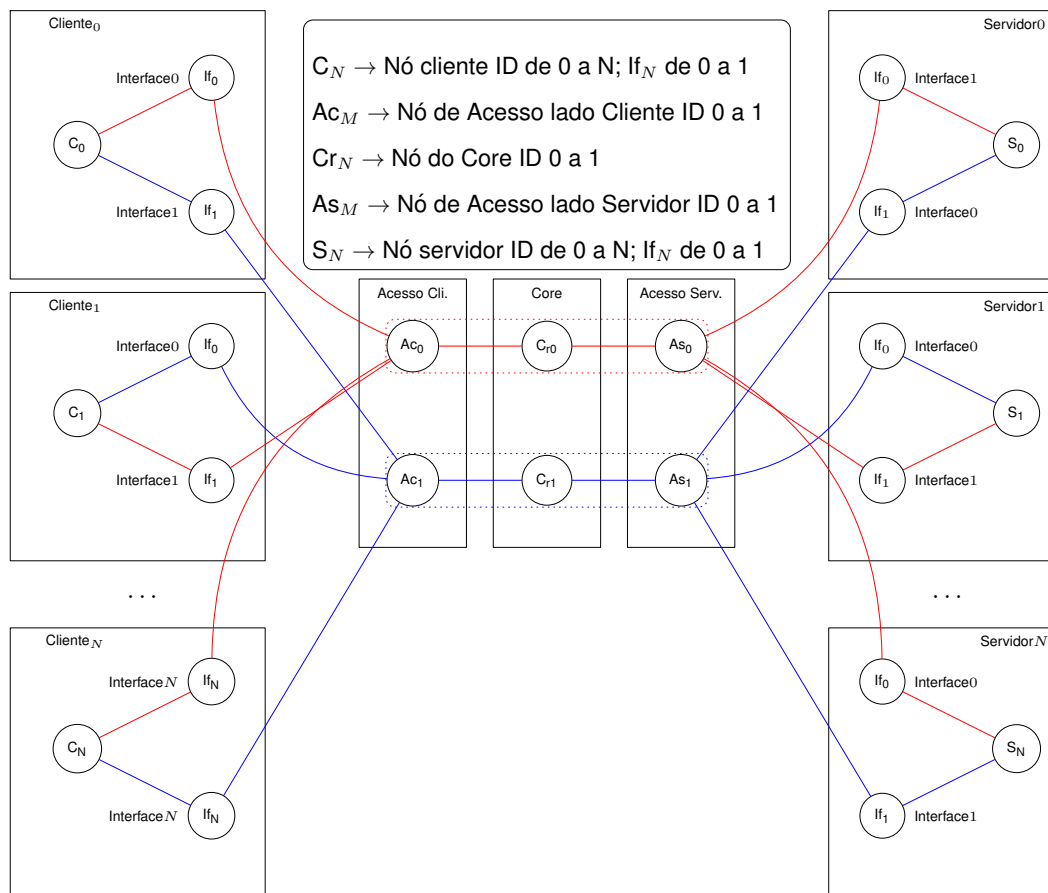


Figura 3.5: Cenário 1: Dois caminhos completamente disjuntos

### 3.3.2 Cenário 2: Três caminhos completamente disjuntos

A figura 3.6 apresenta o modelo de cenário de simulação constituído de três caminhos disjuntos. Este cenário tem a mesma estrutura do cenário 1, mostrado na figura 3.5, com a adição de mais um terceiro caminho identificado pela cor verde. Este terceiro caminho é composto pelos nós  $Ac_2$ ,  $Cr_2$  e  $As_2$  e a este caminho estão conectadas a interface 2 do cliente 0 e a interface 2 do cliente 1.

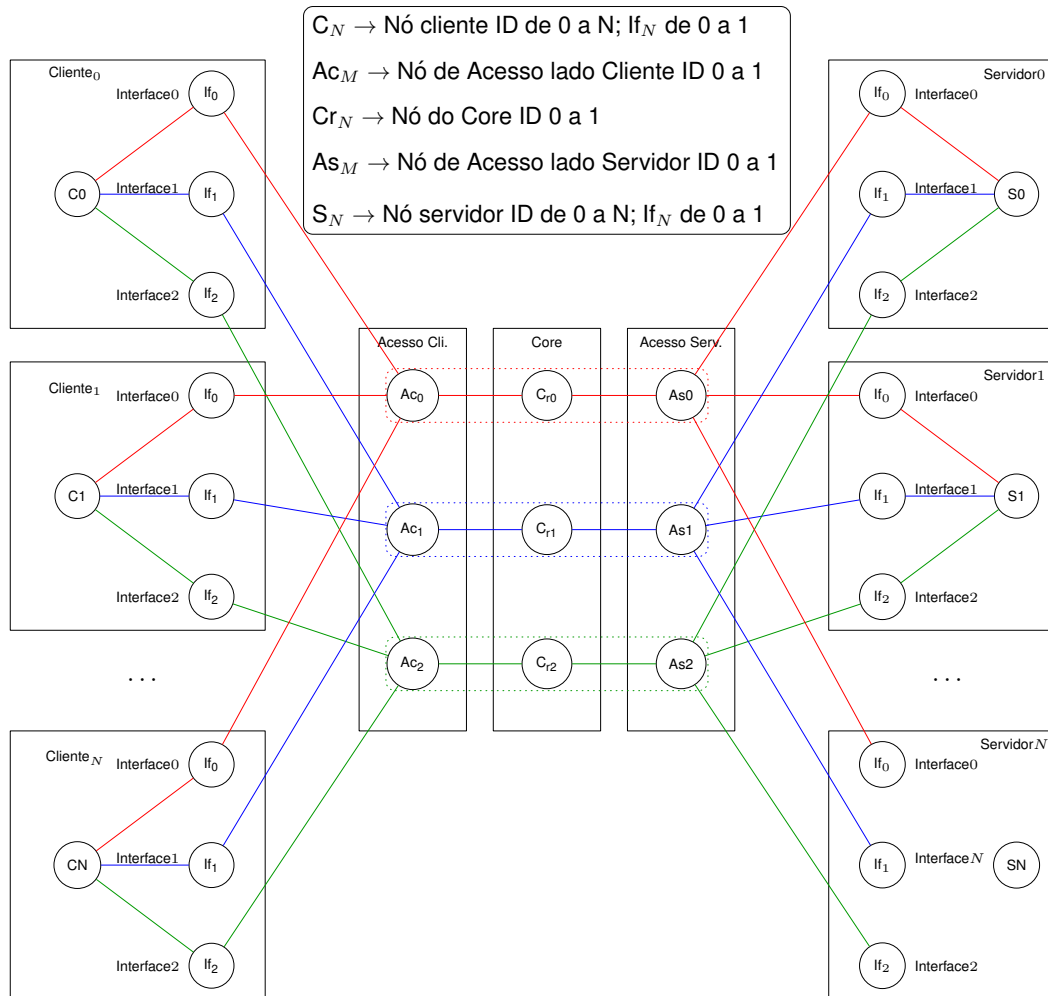


Figura 3.6: Cenário 2: Três caminhos completamente disjuntos

### 3.3.3 Cenário 3: Três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core

A figura 3.7 apresenta o modelo de cenário de simulação constituído de três caminhos, com dois caminhos compartilhando o nó de core. Este cenário tem a mesma estrutura do cenário 2, mostrado na figura 3.6, entretanto o nó do core  $Cr_0$  é compartilhado entre os caminhos de

cor vermelha e verde.

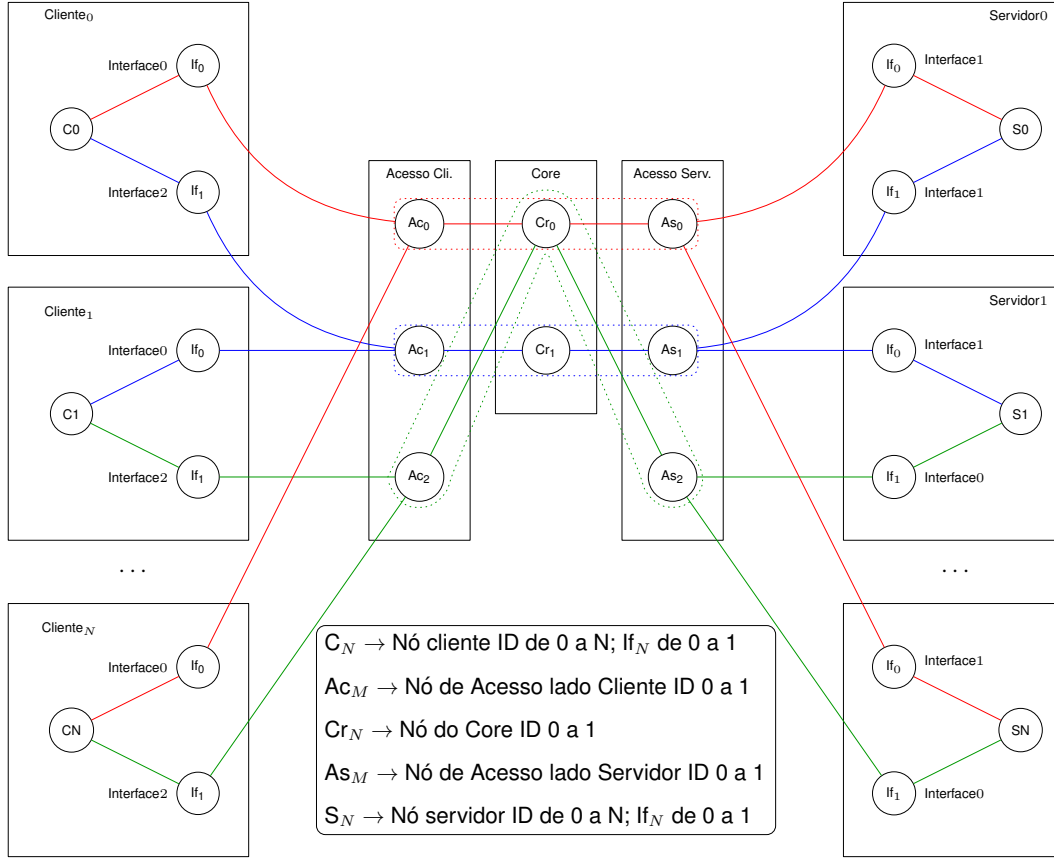


Figura 3.7: Cenário 3: Três caminhos, dois não disjuntos

Conforme a equação 2.8 discutida na sessão 2.2, podemos calcular a diversidade de caminhos neste cenário, tomando como base que:

- O caminho  $P_0$  (cor vermelha) é composto pelos nós:  $Ac_0$ ,  $Cr_0$  e  $As_0$  e pelos links  $L_{Ac_0, Cr_0}$  e  $L_{Cr_0, As_0}$ .
- O caminho  $P_1$  (cor azul) é composto pelos nós:  $Ac_1$ ,  $Cr_1$  e  $As_1$  e pelos links  $L_{Ac_1, Cr_1}$  e  $L_{Cr_1, As_1}$ .
- O caminho  $P_2$  (cor verde) é composto pelos nós:  $Ac_2$ ,  $Cr_0$  e  $As_2$  e pelos links  $L_{Ac_2, Cr_0}$  e  $L_{Cr_0, As_2}$ .

Assim, verifica-se que a diversidade entre  $P_0$  e  $P_2$  é:

$$D(P_2) = 1 - \frac{|\{Ac_2; Cr_0; As_2; L_{Ac_2, Cr_0}; L_{Cr_0, As_2}\} \cap \{Ac_0; Cr_0; As_0; L_{Ac_0, Cr_0}; L_{Cr_0, As_0}\}|}{|\{Ac_0; Cr_0; As_0; L_{Ac_0, Cr_0}; L_{Cr_0, As_0}\}|} \quad (3.3)$$



$$D(P_2) = 1 - \frac{|\{Cr_0\}|}{|\{Ac_0; Cr_0; As_0; L_{Ac_0, Cr_0}; L_{Cr_0, As_0}\}|} \quad (3.4)$$

$$D(P_2) = 1 - \frac{1}{5} = \frac{4}{5} = 0.8 \quad (3.5)$$

Utilizando a mesma equação, verifica-se que  $D(P_1) = 1$  por não ter nenhum nó ou *link* em comum entre  $P_0$  e  $P_1$ . Da mesma forma, constata-se que  $D(P_1, P_2) = 1$  por não existirem nós ou *links* comuns entre  $P_1$  e  $P_2$ .

## CAPÍTULO 4

### RESULTADOS E DISCUSSÃO

A estabilidade de protocolos multicaminho têm por princípio a manutenção de níveis de qualidade de serviço durante toda a transmissão de dados. O aumento do atraso médio, baixos níveis de justiça na divisão do *Throughput* entre os fluxos, queda do *Goodput* são indicadores de que os sistema não se manteve estável durante a transmissão. As métricas apresentadas na sessão 3 são utilizadas no estudo da estabilidade do sistema.

Conforme relatado na sessão 3, as simulações utilizaram o vídeos Aspen, compactado utilizando o codec H.264, com GOP (15,3) e GOP(60,3), e foram utilizados cenários com dois caminhos disjuntos (cenário 1), com três caminhos disjuntos (cenário 2) e com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core (cenário 3), conforme descrito nas sessões 3.3.1, 3.3.2 e 3.3.3 respectivamente. Foram então gerados, para cada cenário e cada um dos protocolos, arquivos com a quantidade de clientes para uma determinada ocupação, iniciando com no mínimo dois agentes para o cenário 1 e no mínimo 3 agentes para o cenário 2 e 3. Quanto maior a quantidade de clientes, maior a ocupação da rede, que foi calculada tendo em vista a quantidade de clientes multiplicado pela taxa média do vídeo a ser transmitido e dividindo-se este resultado por 200 Mbps no caso do cenário 1 e 300 Mbps para o cenário 2 e 3, que equivale ao somatório das taxas dos enlaces entre os nós de acesso e o core da rede em cada um dos cenários.

O tempo estipulado para execução das simulações foi de aproximadamente 470 segundos, para cada um dos níveis de ocupação em cada um dos protocolos, sendo cada simulação repetida por dez vezes. Na análise dos dados foram desconsiderados os dados da simulação dos primeiros 100 segundos, para fins de estabilização dos agentes, e acima de 400 segundos, para se ter uma quantidade constante de dados, por conta do início aleatório dos fluxos entre o segundo 0 e 10.

Inicialmente foram verificados os níveis de atraso médio por ocupação, utilizando o vídeo Aspen, com GOP (60,3) e GOP (15,2). Esses dados são mostrados na figura 4.1, sendo os

gráficos **(a)** e **(b)** para o cenário com dois caminhos disjuntos (cenário 1), conforme descrito na sessão 3.3.1, os gráficos **(c)** e **(d)** para o cenário com três caminhos disjuntos (cenário 2), conforme descrito na sessão 3.3.2 e os gráficos **(e)** e **(f)** para o cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core (cenário 3), conforme descrito na sessão 3.3.3. Os gráficos apresentados na coluna da esquerda [**(a)**,**(c)** e **(e)**] mostram o atraso médio por ocupação para o vídeo com GOP (60,3). Os gráficos da coluna da direita [**(b)**,**(d)** e **(f)**] mostram o atraso médio por ocupação para o vídeo com GOP (15,2).

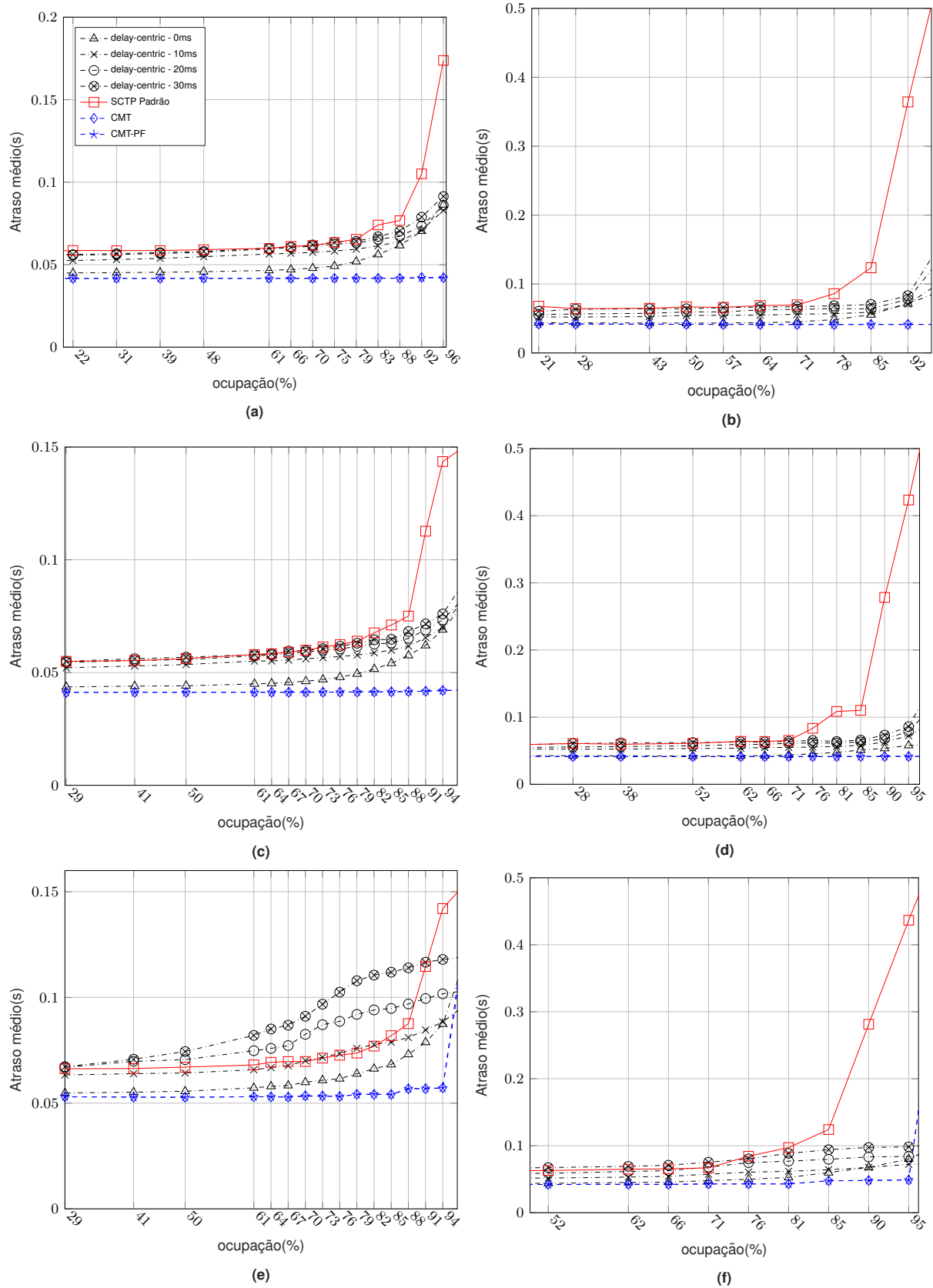


Figura 4.1: Atraso médio vs. ocupação em transmissão de vídeo, sendo os gráficos (a),(c) e (e) o vídeo com GOP (60,3) e os gráficos (b),(d),(f) o vídeo com GOP (15,2). Os gráficos (a) e (b) representam o cenário com dois caminhos disjuntos (cenário 1), os gráficos (c) e (d) o cenário com três caminhos disjuntos (cenário 2) e os gráficos (e) e (f) o cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core (cenário 3). A legenda do gráfico (a) se aplica aos demais gráficos desta figura.

Nos gráficos da figura 4.1, a linha identificada como *delay-centric* - 0ms identifica os dados da simulação do *delay-centric* sem histerese. Já a linha identificada por *delay-centric* - 10ms mostra os dados da simulação do *delay-centric* com 10ms de histerese, a linha identificada por *delay-centric* - 20ms mostra os dados da simulação do *delay-centric* com 20ms de histerese e a linha identificada por *delay-centric* - 30ms mostra os dados da simulação do *delay-centric* com 30ms de histerese.

Na figura 4.1, observa-se um comportamento semelhante para dois e três caminhos disjuntos, tanto para o vídeo com GOP(60,3) apresentado nos gráficos **(a)** e **(c)**, quanto para o vídeo com GOP(15,2) apresentado nos gráficos **(b)** e **(d)**. Esta similaridade dos cenários é justificada por conta de os caminhos serem totalmente disjuntos. Assim, a degradação de desempenho em um dos caminhos não afeta os demais.

O gráfico **(e)** da figura 4.1, é o que mostra um aumento do atraso médio do *delay-centric* com histerese de 20ms e 30ms em ocupações maiores que 29%. O SCTP Padrão e o *delay-centric*, sem histerese e com histerese de 10ms, apresentaram um aumento no atraso médio para ocupações maiores que 70%. O CMT e CMT-PF não apresentaram variações no nível de atraso nos gráficos **(a),(b),(c),(d)**, mas no gráfico **(e)** ambos apresentaram aumento no atraso médio em ocupação a partir de 91%. No gráfico **(f)** ambos apresentaram aumento no atraso médio em ocupação a partir de 91%. Nos resultados obtidos, observa-se maiores níveis de atraso médio para o SCTP padrão durante a transmissão do vídeo.

A fim de complementar a investigação do comportamento do atraso mostrado dos gráficos da figura 4.1, foram analisados os dados do *Goodput*, apresentados na figura 4.2. Nela, os gráficos **(a)** e **(b)** representam o cenário com dois caminhos disjuntos (cenário 1), conforme descrito na sessão 3.3.1, os gráficos **(c)** e **(d)** para o cenário com três caminhos disjuntos (cenário 2), conforme descrito na sessão 3.3.2 e os gráficos **(e)** e **(f)** para o cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core (cenário 3), conforme descrito na sessão 3.3.3. Os gráficos apresentados na coluna da esquerda **[(a),(c) e (e)]** mostram o atraso médio por ocupação para o vídeo com GOP (60,3). Os gráficos da coluna da direita **[(b),(d) e (f)]** mostram o atraso médio por ocupação para o vídeo com GOP (15,2).

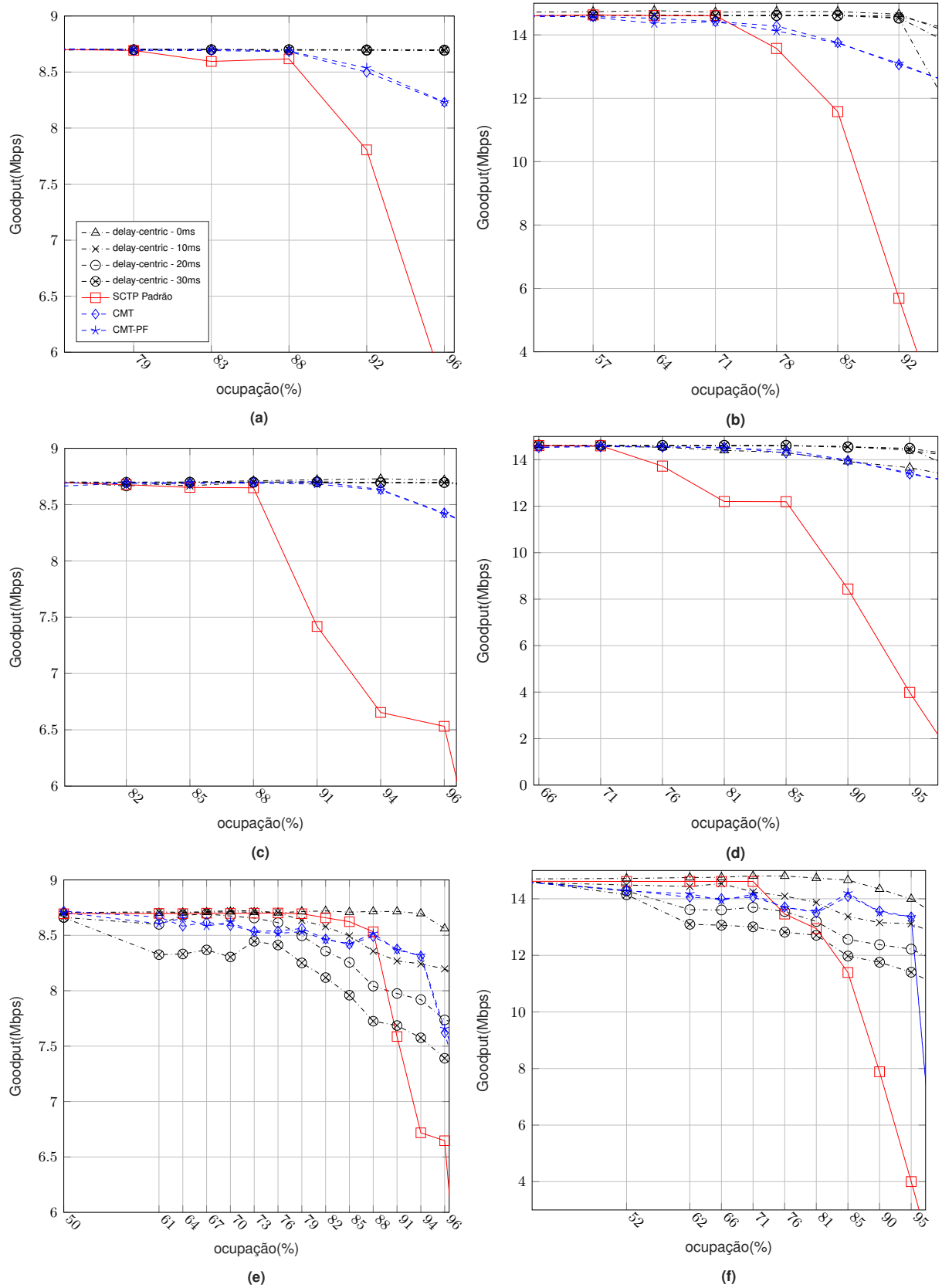


Figura 4.2: Goodput médio vs. ocupação em transmissão de vídeo, sendo a figura (a) o vídeo com GOP (60,3) e a figura (b) o vídeo com GOP (15,2). Os gráficos (a) e (b) representam o cenário com dois caminhos disjuntos (cenário 1), os gráficos (c) e (d) o cenário com três caminhos disjuntos (cenário 2) e os gráficos (e) e (f) o cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core (cenário 3). A legenda do gráfico (a) se aplica às demais gráficos desta figura.

Nos gráficos da figura 4.2, a linha identificada como *delay-centric* - 0ms identifica os dados da simulação do *delay-centric* sem histerese. Já a linha identificada por *delay-centric* - 10ms mostra os dados da simulação do *delay-centric* com 10ms de histerese, a linha identificada por *delay-centric* - 20ms mostra os dados da simulação do *delay-centric* com 20ms de histerese e a linha identificada por *delay-centric* - 30ms mostra os dados da simulação do *delay-centric* com 30ms de histerese.

Na figura 4.2 observa-se queda no *Goodput* do SCTP padrão para ocupações maiores que 71% nos gráficos **(b)**, **(d)** e **(f)**. Já os gráficos **(a)** e **(e)** observa-se redução do *Goodput* em ocupações maiores que 79%. No gráfico **(c)** a redução do *Goodput* se dá a partir de 82% de ocupação. O CMT e o CMT-PF apresentaram degradação do *Goodput* em ocupações maiores que: 88% no gráfico **(a)**, 57% no gráfico **(b)**, 91% no gráfico **(c)**, 76% no gráfico **(d)** e 50% nos gráficos **(e)** e **(f)**.

O *delay-centric* não apresentou queda no *Goodput*, independente da histerese utilizada nos gráficos **(a)** e **(c)** da figura 4.2, mas apresentou redução do *Goodput* em ocupações maiores que 92% no gráfico **(b)** e em ocupações maiores que 95% no gráfico **(d)**. No gráfico **(e)** o *delay-centric* com histerese de 30ms, apresentou degradação do *Goodput* em ocupações maiores que 49%, o *delay-centric* com histerese de 20ms apresentou degradação do *Goodput* em ocupações maiores que 70%, o *delay-centric* com histerese de 10ms apresentou degradação do *Goodput* em ocupações maiores que 76% e o *delay-centric* sem histerese, apresentou degradação do *Goodput* em ocupações maiores que 94%. No gráfico **(f)**, o *delay-centric* apresentou melhor *Goodput* quando não foi utilizado histerese, com redução do *Goodput* em ocupações maiores que 85%. Quando utilizado histerese de 10ms, o *delay-centric* apresentou queda do *Goodput* em ocupações maiores que 66%. Com 20ms e 30ms de histerese, houve redução do *Goodput* em ocupações maiores que 52%.

O CMT e o CMT-PF apresentaram redução do *Goodput* em ocupações maiores que: 88% no gráfico **(a)**, 57% no gráfico **(b)**, 91% no gráfico **(c)**, 76% no gráfico **(d)** e 50% nos gráficos **(e)** e **(f)**.

O *Goodput* foi calculado utilizando os pacotes que chegaram ao destino em tempo de serem decodificados. Ao analisar o *Goodput* em relação as diferentes técnicas, observa-se que a queda do *Goodput* do SCTP Padrão e do *delay-centric*, independente da histerese

utilizada, observados nos gráficos da figura 4.2, é justificada pelo aumento do atraso médio observado nos gráficos da figura 4.1.

Na figura 4.3 são apresentados os gráficos de *Throughput* em relação à ocupação sendo o gráfico **(a)** corresponde ao vídeo com GOP (60,3) e o gráfico **(b)** o vídeo com GOP(15,2), para o cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core. Os gráficos dos demais cenários apresentaram resultados em conformidade com os gráficos da figura 4.2.

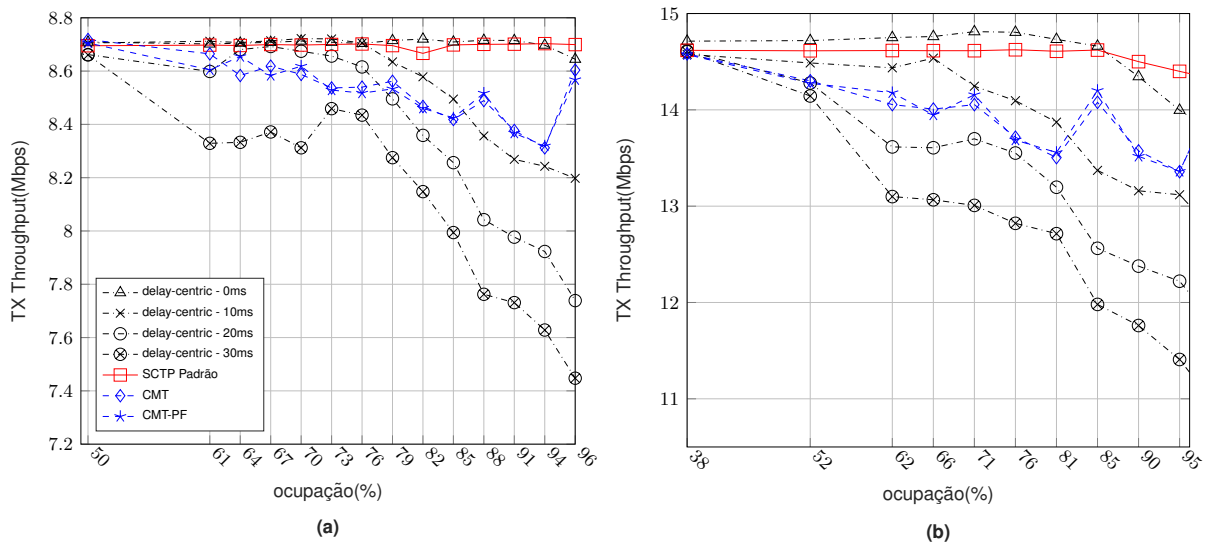


Figura 4.3: Throughput médio vs. ocupação em transmissão de vídeo, sendo a figura **(a)** o vídeo com GOP (60,3) e a figura **(b)** o vídeo com GOP (15,2). Os gráficos **(a)** e **(b)** representam o cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core (cenário 3). A legenda do gráfico **(a)** se aplica aos demais gráficos desta figura.

No gráfico **(a)** da figura 4.3, o SCTP padrão apresentou pequena degradação do *Throughput* somente em 82% de ocupação. O *delay-centric* sem histerese apresentou pequena redução do *Throughput* em ocupações maiores que 94%. Quando utilizado histerese de 10ms, o *delay-centric* apresentou queda do *Throughput* em ocupações maiores que 76%. Quando utilizada histerese de 20ms, o *delay-centric* apresentou queda do *Throughput* em ocupações maiores que 67%, e quando utilizada histerese de 30ms, o *delay-centric* apresentou queda do *Throughput* em ocupações maiores que 50%. O CMT e o CMT-PF por sua vez, apresentaram redução do *Throughput* em ocupações maiores que 50%.

No gráfico **(b)** da figura 4.3, verifica-se que tanto o SCTP padrão quanto o *delay-centric* sem histerese, reduzem o *Throughput* apenas em ocupações maiores que 85%. Já o *delay-*



*centric* com histerese de 10ms reduz o *Throughput* em ocupações maiores que 66%. O CMT, CMT-PF e *delay-centric* com histerese de 20ms e 30ms, reduziram o *Throughput* em ocupações maiores que 38%.

A atuação do algoritmo de controle de congestionamento, conforme elucidado na sessão 2.1.1.6, é evidenciada com a redução da taxa de envio de pacotes pelo nó de origem com a finalidade de se reduzir o congestionamento da rede como um todo. Assim podemos concluir que a redução do *Goodput* pelo CMT e CMT-PF observados na figura 4.2 em ocupações maiores que: 88% no gráfico (a), 57% no gráfico (b), 91% no gráfico (c), 76% no gráfico (d) e 50% nos gráficos (e) e (f), são resultantes da redução da taxa de envio de pacotes por parte do *host* de origem, mostrando a atuação do mecanismo de controle de congestionamento por parte do CMT e do CMT-PF.

O índice de justiça na divisão do *Throughput* mostra o balanceamento do tráfego entre todos os agentes que competem pelo recurso. Essa métrica é apresentada na figura 4.4 sendo o gráfico (a) corresponde ao vídeo com GOP (60,3) e o gráfico (b) o vídeo com GOP(15,2), ambos para o cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core.

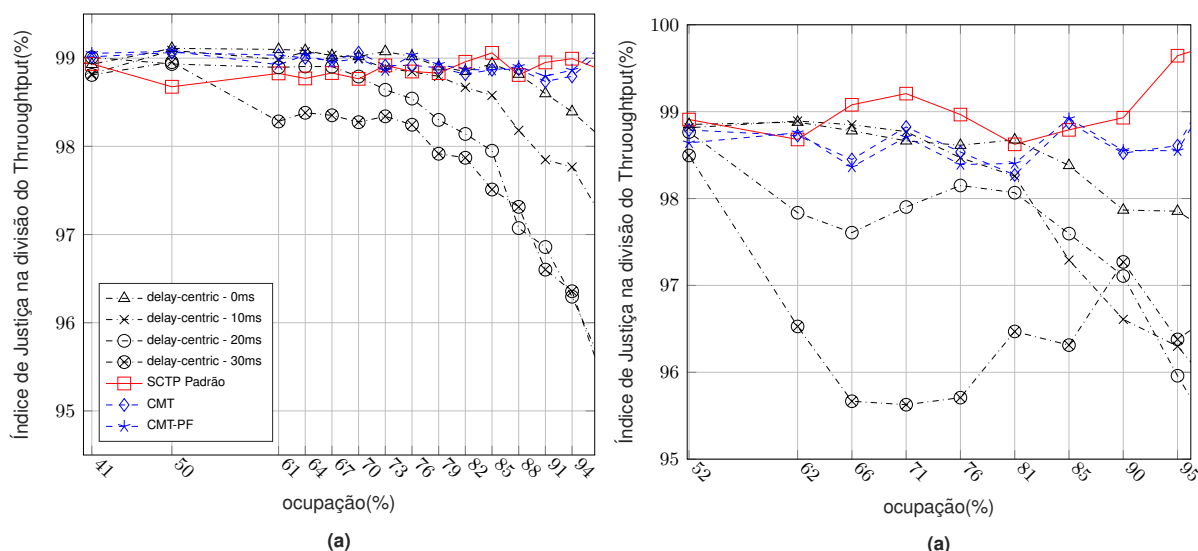


Figura 4.4: Índice de justiça de Jain médio vs. ocupação em transmissão de vídeo, sendo a figura (a) o vídeo com GOP (60,3) e a figura (b) o vídeo com GOP (15,2). Os gráficos (a) e (b) representam o cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core (cenário 3). A legenda do gráfico (a) se aplica aos demais gráficos desta figura.

Nos gráficos (a) e (b) da figura 4.4, o Sctp Padrão, o CMT e o CMT-PF não apresentou

redução significativa do índice de justiça de divisão do *Throughput*, ambos variando em torno de 90%. Observa-se em todos os gráficos da figura 4.1 um aumento no atraso médio dos pacotes enviados utilizando o SCTP padrão, que se dá por conta do congestionamento que se estabelece na rede. Este aumento no atraso para entrega de pacotes na rede se dá pela formação de fila nos roteadores do núcleo da rede e mostra que, apesar do congestionamento, o SCTP padrão não reduziu sua taxa de envio. Por não reduzir sua taxa de envio de pacotes, é esperado que não haja redução na justiça na divisão do *Throughput*, o que justifica os valores apontados no gráfico.

A redução do *Throughput* pelo CMT e CMT-PF em ocupações maiores que 38%, apresentados na figura 4.3 gráficos **(e)** e **(f)** (correspondentes ao cenário 3), mostram que a atuação do mecanismo de controle de congestionamento do CMT e CMT-PF está funcionando conforme esperado. Na tentativa de minimizar o congestionamento da rede como um todo, os fluxos reduzem sua taxa de envio de pacotes de maneira igualitária. Por isto observa-se que não há aumento do atraso médio de pacotes por parte do CMT e CMT-PF, conforme pode ser observado nos gráficos **(e)** e **(f)** (correspondentes ao cenário 3) da figura 4.1, mesmo com o aumento da ocupação.

Observa-se ainda que no gráfico **(a)** da figura 4.4, o *delay-centric* sem histerese apresentou redução do índice de justiça de divisão do *Throughput* em ocupações maiores que 88%. O *delay-centric* com histerese de 10ms, apresentou redução do índice de justiça de divisão do *Throughput* em ocupações maiores que 79%. Já o *delay-centric* com histerese de 20ms, apresentou redução do índice de justiça de divisão do *Throughput* em ocupações maiores que 67% e o *delay-centric* com histerese de 30ms apresentou redução do índice de justiça de divisão do *Throughput* em ocupações maiores que 50%.

Ainda na figura 4.4, no gráfico **(b)** figura 4.4, observa-se que a justiça na divisão do *Throughput* para o *delay-centric* sem histerese e com histerese de 10ms, apresenta redução do índice de justiça de divisão do *Throughput* em ocupações maiores que 81%. Já o *delay-centric* com histerese de 20ms e 30ms, apresentaram redução do índice de justiça de divisão do *Throughput* em ocupações maiores que 52%.

O comportamento na justiça na divisão do *Throughput* tanto para o *delay-centric* sem histerese quanto para o *delay-centric* com histerese de 10ms mostra que algoritmos de escolha

de rota mais agressivos conseguem obter vantagem quanto em situações de congestionamento na rede. Conforme apresentado na sessão 2.1.1.7, Jungmaier, Rathgeb e Tüxen [24] mostrou que a redução do valor da variável PMR, que leva o SCTP a efetuar a troca de caminho na presença da primeira perda de pacote, aumenta a performance em cenários de congestionamento de rede onde tipicamente ocorrem grandes variações de atraso médio e perda de pacotes. Já Grinnemo e Brunstrom [25] mostrou que grandes buffers em roteadores podem causar grandes variações de atraso, o que tende a aumentar o RTO e consequentemente o SRTT daquele caminho. Aliando as ideias de ambos, podemos verificar que no cenário 3 temos três caminhos sendo dois compartilhando um nó no núcleo da rede, o que leva a uma situação de variação no atraso médio dos pacotes dado o compartilhamento do nó. Assim confirma-se que o delay-centric sem histerese ou com histerese de 10ms por serem mais agressivos obtiveram maior vantagem em relação aos demais protocolos, justamente por efetuarem a troca de caminho ao perceberem uma mínima variação no atraso.

A quantidade de trocas de caminho pelo delay-centric pode ser verificada na figura 4.5. No gráfico **(a)** da figura 4.5 observa-se que o delay-centric apresenta uma quantidade de trocas de caminho média de aproximadamente 80 vezes durante toda a transmissão, tendo um incremento quase próximo de linear da quantidade de trocas conforme se aumenta a ocupação. Já na figura **(b)** o delay-centric sem histerese apresenta uma quantidade de trocas de caminho média de aproximadamente 140 vezes durante toda a transmissão, apresentando uma redução na quantidade média de trocas de caminho até uma ocupação de 38% para aproximadamente 110 trocas e apresentando um incremento próximo de linear a partir de 52% de ocupação.

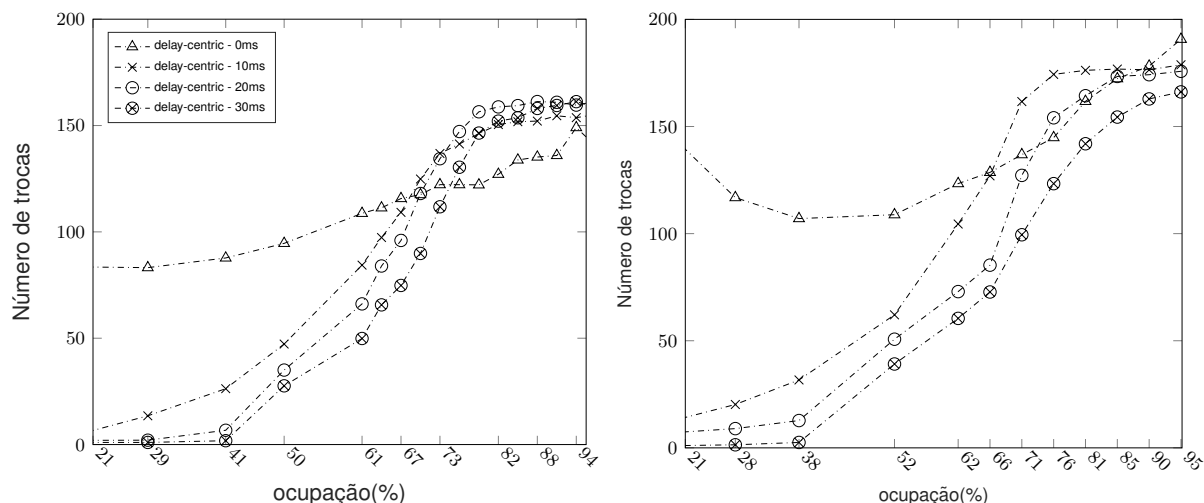


Figura 4.5: Número médio de trocas de caminho vs. ocupação em transmissão de vídeo, sendo a figura **(a)** o vídeo com GOP (60,3) e a figura **(b)** o vídeo com GOP (15,2). Os gráficos **(a)** e **(b)** representam o cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core (cenário 3). A legenda do gráfico **(a)** se aplica aos demais gráficos desta figura.

Ainda sobre a figura 4.5, em ambos os gráficos observa-se que o delay-centric que possui histerese de 10ms, 20ms e 30ms apresentam um comportamento semelhante no que tange a quantidade média de trocas de caminho. Neste caso, tanto no gráfico **(a)** quanto no gráfico **(b)** observa-se que em baixas ocupações a quantidade média de trocas de caminho fica próxima de zero em baixas ocupações e apresenta crescimento exponencial com o aumento da ocupação da rede.

Uma grande quantidade de trocas de caminho pode levar o sistema a apresentar um efeito manada, conforme descrito por Leung et al. [18], podendo levar a situações de grande variação no atraso médio dos pacotes. Nesta condição, uma elevada quantidade de trocas de caminho pode levar a grandes variações no atraso fim-a-fim dos pacotes, dada rajadas de pacotes que ocupariam as filas dos roteadores no núcleo de rede desordenadamente, prejudicando o sistema como um todo. Nas simulações realizadas não ocorreram situações onde uma grande quantidade de trocas de caminho em baixas ocupações levasse a um efeito manada e fosse prejudicial à rede como um todo. Mas, dadas as limitações do modelo esta possibilidade não pode ser descartada.

Considerando a transmissão de vídeo utilizando o SCTP Padrão e o CMT, os resultados obtidos neste trabalho comprovam os dados obtidos por Xu et al. [7], onde o SCTP Padrão por conta de suas limitações quanto à troca de caminho, conforme discutido na sessão 2.1.1.7,

continua enviando pacotes no caminho congestionado até que seja evidenciada uma perda de pacotes. No caso do sistema simulado não ocorreram perda de pacotes e a transmissão foi configurada como não confiável o que fez com que o SCTP Padrão não realizasse a troca de caminho em nenhum momento. O SCTP Padrão mostrou melhor desempenho que o CMT, CMT-PF e *delay-centric* com histerese de 20 ms e 30 ms, no cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core, principalmente devido à dificuldade de lidar com variações no atraso médio de pacotes por parte do CMT e do CMT-PF, conforme descrito na sessão 2.1.1.9.

O *delay-centric* com histerese de 10 milissegundos, 20 milissegundos e 30 milissegundos, apresentaram melhor performance que o SCTP padrão, CMT e CMT-PF nos cenários com dois e três caminhos disjuntos. No cenário com duas redes de acesso compartilhando um core, o *delay-centric* com histerese de 10 milissegundos apresentou desempenho igual ou superior ao CMT e CMT-PF e o *delay-centric* com histerese de 20 milissegundos e 30 milissegundos apresentaram desempenho inferior ao CMT e o CMT-PF.

Analisando as figuras 4.1, 4.2, 4.3 e 4.5 podemos verificar que *delay-centric* sem histerese foi o protocolo que apresentou melhor performance para transmissão de vídeos em tempo real, mantendo boa relação entre o *Throughput*, *Goodput*, justiça na divisão do *Throughput* e baixo atraso médio dos pacotes. Os resultados obtidos para os cenários com dois e três caminhos completamente disjuntos comprovam os dados obtidos por Kelly et al. [14] e Leung et al. [18], onde o *delay-centric* se mostrou o mais eficaz para transmissão de dados que o SCTP padrão. Já os resultados obtidos para o *delay-centric* sem histerese no cenário cenário com duas redes de acesso compartilhando um core, com duas redes de acesso compartilhando um core, vêm a complementar os resultados obtidos por Kelly et al. [14] e Leung et al. [18], uma vez que estes autores se limitaram ao estudo de cenários com dois caminhos disjuntos e tráfegos de voz.

## CAPÍTULO 5

### CONCLUSÃO

Com o aumento da diversidade dos meios de acesso à internet, surgiu a possibilidade de incorporar múltiplas interfaces de rede em um único dispositivo. Neste contexto foi proposto o protocolo SCTP, que pode fazer uso das múltiplas interfaces de rede para prover redundância ou balanceamento de carga. O *delay-centric* foi inicialmente proposto para realizar *handover* entre redes móveis no protocolo SCTP. Sua proposta de realizar a troca de caminho utilizando o valor do atraso médio suavizado mostrou uma melhora significativa quando utilizado em aplicações de tempo-real. Vários autores tem estudado a utilização do SCTP padrão, do CMT e do CMT-PF em transmissões de vídeo de tempo real, dado que este tipo de tráfego aumentou significativamente nos últimos anos.

Este trabalho se propôs a estudar a performance de transmissões de vídeo HD com diferentes características de GOP, por protocolos de transporte multicaminho. Simulações foram realizadas com vídeos de características distintas e foram analisados para fins de comparação os comportamentos do *delay-centric* com diferentes tempos de guarda, SCTP padrão, CMT e do CMT-PF. Foram utilizados cenários com vários níveis de diversidade e observadas métricas de atraso médio, *Goodput* e justiça na divisão do *Throughput* dos agentes que enviavam o vídeo.

Os resultados da simulação mostram que *delay-centric* sem histerese foi o protocolo que apresentou melhor desempenho na transmissão de vídeo streaming, mantendo boa relação entre o *throughput* e o atraso médio. A medida que se aumentou a histerese, o atraso médio aumentou, o *Throughput* foi reduzido e a justiça na divisão do *throughput* piorou. No entanto, observa-se que o *Goodput* na transmissão do vídeo foi sempre melhor para o *delay-centric*.

Nos cenários com dois e três caminhos completamente disjuntos, o *delay-centric*, independente da histerese, teve desempenho superior em relação aos demais protocolos estudados. O CMT e o CMT-PF tiveram o mesmo comportamento relatado na literatura, mostrando melhor desempenho que o SCTP Padrão. O SCTP Padrão não se beneficiou da existência de

múltiplos caminhos pois este somente realiza a troca de caminhos após detecção de perdas sucessivas de pacotes, apresentando assim o pior desempenho entre todos os protocolos estudados nos cenários com dois e três caminhos completamente disjuntos.

No cenário com três nós de acesso interligados a dois cores, com duas redes de acesso compartilhando um core, os resultados indicaram que o *delay-centric* sem histerese obteve o melhor desempenho dentre todos os protocolos estudados. O SCTP padrão obteve melhor performance que o CMT e o CMT-PF. O CMT e o CMT-PF apresentaram desempenho semelhante ao *delay-centric* com histerese de 10 ms, e superior ao *delay-centric* com histerese de 20 ms e 30 ms.

Os resultados obtidos por este trabalho vêm a complementar os estudos sobre o SCTP, comparando diferentes implementações do protocolo e mostrando as vantagens de sua utilização em dispositivos com múltiplas interfaces de rede. Assim, como sugestão de continuidade deste trabalho, indica-se a análise do desempenho dos mesmos protocolos em ambientes com perdas de pacotes, podendo ser estendida para novas versões do CMT, para avaliar o comportamento do mecanismo de controle de erros e congestionamento. Neste caso, o número de trocas impactaria fortemente no desempenho do *delay-centric*.

## BIBLIOGRAFIA

- [1] STERBENZ, J. P.; HUTCHISON, D.; ÇETINKAYA, E. K.; JABBAR, A.; ROHRER, J. P.; SCHÖLLER, M.; SMITH, P. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. **Computer Networks**, v. 54, n. 8, p. 1245 – 1265, 2010.
- [2] CLARK, D. The design philosophy of the darpa internet protocols. **SIGCOMM Computer Communication Review**, v. 18, n. 4, Agosto 1988.
- [3] MEYER, J. F. Performability evaluation: where it is and what lies ahead. In: **Proceedings of 1995 IEEE International Computer Performance and Dependability Symposium**. 1995. p. 334–343.
- [4] KVALBEIN, A.; HANSEN, A. F.; CICIC, T.; GJESSING, S.; LYSNE, O. Multiple routing configurations for fast ip network recovery. **IEEE/ACM Transactions on Networking**, v. 17, n. 2, p. 473–486, Abril 2009.
- [5] STEWART, R. **Stream Control Transmission Protocol - RFC 4960**. IETF, Setembro 2007. Request for Comments, 4960. RFC 4960 (Proposed Standard).
- [6] ROHRER, J. P. **End-to-End Resilience Mechanisms for Network Transport Protocols**. Tese (Doutorado) — The University of Kansas, Lawrence, KS, Novembro 2011.
- [7] XU, C.; FALLON, E.; QIAO, Y.; ZHONG, L.; MUNTEAN, G. M. Performance evaluation of multimedia content distribution over multi-homed wireless networks. **IEEE Transactions on Broadcasting**, v. 57, n. 2, p. 204–215, Junho 2011.
- [8] ZIMMERMANN, A.; EDDY, W.; EGGERT, L. **Moving Outdated TCP Extensions and TCP-Related Documents to Historic or Informational Status - RFC 7805**. Abril 2016. Request for Comments, 7805. RFC 7805 (Informational).
- [9] POSTEL, J. **User Datagram Protocol - RFC 768**. IETF, Agosto 1980. RFC 768 (INTERNET STANDARD).
- [10] SCHULZRINNE, H.; CASNER, S.; FREDERICK, R.; JACOBSON, V. **RTP: A Transport Protocol for Real-Time Applications - RFC 3550**. IETF, Julho 2003. Request for Comments, 3550. RFC 3550 (INTERNET STANDARD).



- [11] GREENGRASS, J.; EVANS, J.; BEGEN, A. C. Not all packets are equal, part i: Streaming video coding and SLA requirements. **IEEE Internet Computing**, v. 13, n. 1, p. 70–75, Janeiro 2009.
- [12] FLOYD, S.; FALL, K. Promoting the use of end-to-end congestion control in the internet. **IEEE/ACM Trans. Netw.**, v. 7, n. 4, Agosto 1999.
- [13] HAN, J.; WATSON, D.; JAHANIAN, F. An experimental study of internet path diversity. **IEEE Transactions on Dependable and Secure Computing**, v. 3, n. 4, p. 273–288, Outubro 2006.
- [14] KELLY, A.; MUNTEAN, G.; PERRY, P.; J., M. Delay-centric handover in SCTP over wlan. **Transactions on Automatic Control, Computer Science**, Buletinul Stiintific al Universitatii Politehnica din Timisoara, Universitati Politehnica din Timisoara, ROMANIA, v. 49, p. 211–216, 2004.
- [15] TRAN, V.-H.; CONINCK, Q. D.; HESMANS, B.; SADRE, R.; BONAVENTURE, O. Observing real multipath TCP traffic. **Computer Communications**, v. 94, p. 114 – 122, 2016.
- [16] HUSSEIN, D.; HASSAN, M.; LANDOLSI, T. Performance evaluation of a path diversity-based video streaming approach over best effort networks. In: **2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation**. 2014. p. 536–541.
- [17] POSTEL, J. **Transmission Control Protocol**. IETF, Setembro 1981. Request for Comments, 793. RFC 793 (INTERNET STANDARD).
- [18] LEUNG, V.; RIBEIRO, E.; WAGNER, A.; IYENGAR, J. **Multihomed Communication with SCTP (Stream Control Transmission Protocol)**. CRC Press, 2012. (ENGnetBASE 2015).
- [19] STEWART, R.; XIE, Q.; TUOXEN, M.; MARUYAMA, S.; KOZUKA, M. **Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration**. IETF, Setembro 2007. Request for Comments, 5061. RFC 5061 (Proposed Standard).
- [20] KARN, P.; SIMPSON, W. **Photuris: Session-Key Management Protocol**. IETF, Março 1999. Request for Comments, 2522. RFC 2522 (Experimental).

- [21] ALLMAN, M.; PAXSON, V. On estimating end-to-end network path properties. **SIGCOMM Computer Communication Review**, ACM, New York, NY, USA, v. 29, n. 4, p. 263–274, Agosto 1999.
- [22] STEWART, R.; XIE, Q. **Stream Control Transmission Protocol (SCTP): A Reference Guide**. Addison-Wesley, 2002.
- [23] SCHARF, M.; KIESEL, S. Internet-Draft, **Quantifying Head-of-Line Blocking in TCP and SCTP**. Julho 2013. Working Draft.
- [24] JUNGMAIER, A.; RATHGEB, P. E.; TÜXEN, M. On the use of SCTP in Failover-Scenarios. **Journal of Systemics, Cybernetics, Informatics**, ISS - International Institute of Informatics, Systemics, Julho 2002.
- [25] GRINNEMO, K.-J.; BRUNSTROM, A. Impact of traffic load on SCTP failovers in sigtran. In: **4th International Conference on Networking(ICN 2005), Reunion Island, France, April 17-21, 2005, Proceedings, Part I**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 774–783.
- [26] BUDZISZ, L.; FERRUS, R.; GRINNEMO, K.; BRUNSTROM, A.; CASADEVALL, F. An analytical estimation of the failover time in SCTP multihoming scenarios. In: **Wireless Communications, Networking Conference, 2007.WCNC 2007. IEEE**. 2007. p. 3929–3934.
- [27] CARO, A. L.; AMER, P. D.; STEWART, R. R. End-to-end failover thresholds for transport layer multihoming. **Annals of Telecommunications**, v. 1, n. 1-2, p. 99–105 Vol. 1, Outubro 2004.
- [28] CARO, A. L.; AMER, P. D.; STEWART, R. R. Rethinking end-to-end failover with transport layer multihoming. **Annales Des Télécommunications**, v. 61, n. 1, p. 92–114, 2006.
- [29] NOONAN, J.; PERRY, P.; MURPHY, S.; MURPHY, J. Stall and path monitoring issues in SCTP. In: **Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications**. 2006. p. 1–9.
- [30] NATARAJAN, P.; IYENGAR, J.; AMER, P.; STEWART, R. Concurrent multipath transfer using transport layer multihoming: Performance under network failures. In: **Military Communications Conference, 2006. MILCOM 2006. IEEE**. 2006. p. 1–7.

- [31] STEWART, R.; RAMALHO, M.; XIE, Q.; TUEXEN, M.; CONRAD, P. **Stream Control Transmission Protocol (SCTP) Partial Reliability Extension**. IETF, Maio 2004. Request for Comments, 3758. RFC 3758 (Proposed Standard).
- [32] GOFF, T.; PHATAK, D. S. Unified transport layer support for data striping and host mobility. **IEEE Journal on Selected Areas in Communications**, v. 22, n. 4, p. 737–746, Maio 2004.
- [33] AHMED, A.; SAADAWI, T.; LEE, M. Improving throughput, reliability in mobile wireless networks via transport layer bandwidth aggregation. **Computer Networks**, v. 46, n. 5, p. 635 – 649, 2004.
- [34] AL, A.; SAADAWI, T.; LEE, M. Ls-sctp: a bandwidth aggregation technique for stream control transmission protocol. **Computer Communications**, v. 27, n. 10, p. 1012 – 1024, 2004.
- [35] YE, G.; SAADAWI, T.; LEE, M. IPCC-SCTP: an enhancement to the standard SCTP to support multi-homing efficiently. In: **Performance, Computing, Communications, 2004 IEEE International Conference on**. 2004. p. 523–530.
- [36] IYENGAR, J.; AMER, P.; STEWART, R. Concurrent multipath transfer using transport layer multihoming: performance under varying bandwidth proportions. In: **Military Communications Conference, 2004. MILCOM 2004. 2004 IEEE**. 2004. v. 1, p. 238–244 Vol. 1.
- [37] IYENGAR, J.; AMER, P.; STEWART, R. Concurrent multipath transfer using sctp multihoming over independent end-to-end paths. **Networking, IEEE/ACM Transactions on**, v. 14, n. 5, p. 951–964, Outubro 2006.
- [38] IYENGAR, J.; AMER, P.; STEWART, R. Retransmission policies for concurrent multipath transfer using sctp multihoming. In: **Networks, 2004. (ICON 2004). Proceedings. 12th IEEE International Conference on**. 2004. v. 2, p. 713–719 vol.2.
- [39] IYENGAR, J.; AMER, P.; STEWART, R. Receive buffer blocking in concurrent multipath transfer. In: **Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE**. 2005. v. 1, p. 6.

- [40] IYENGAR, R.; AMER, P.; STEWART, R. Performance implications of a bounded receive buffer in concurrent multipath transfer. **Computer Communications**, v. 30, n. 4, p. 818 – 829, 2007.
- [41] NISHIDA, Y.; NATARAJAN, P.; CARO, A.; AMER, P.; NIELSEN, K. **SCTP-PF: A Quick Failover Algorithm for the Stream Control Transmission Protocol**. IETF, Abril 2016. Request for Comments, 7829. RFC 7829 (Proposed Standard).
- [42] PREETHI, N.; NASIF, E.; PAUL, A.; STEWART, R. Concurrent multipath transfer during path failure. **Computer Communications**, v. 32, n. 15, p. 1577 – 1587, 2009.
- [43] ROHRER, J.; NAIDU, R.; STERBENZ, J. Multipath at the transport layer: An end-to-end resilience mechanism. In: **Ultra Modern Telecommunications Workshops, 2009. ICUMT '09. International Conference on**. 2009. p. 1–7.
- [44] ROHRER, J. P.; JABBAR, A.; STERBENZ, J. P. G. Path diversification: A multipath resilience mechanism. In: **Proceedings of the IEEE 7th International Workshop on the Design of Reliable Communication Networks (DRCN)**. Washington, DC, USA: , 2009. p. 343–351.
- [45] Federal Communications Commission (FCC). **Measuring Broadband America Fixed Broadband Report**. Disponível em: <<https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-broadband-america-2015>>.
- [46] European Commission (EC). **Broadband access in the EU - Broadband indicators - Janeiro 2016**. Disponível em: <<https://ec.europa.eu/digital-single-market/en/news/broadband-access-eu-data-january-2016>>.
- [47] ITU-T. **RECOMMENDATION H.264**. International Telecommunication Union: Place des Nations, Geneva, Switzerland, Fevereiro 2014.
- [48] Digital Subscriber Line(DSL)-FORUM. **Triple-play Services Quality of Experience (QoE) Requirements**. Dezembro 2006.
- [49] SIKORA, T. MPEG digital video-coding standards. **IEEE Signal Processing Magazine**, v. 14, n. 5, p. 82–100, Setembro 1997.

[50] GREENGRASS, J.; EVANS, J.; BEGEN, A. C. Not all packets are equal, part i: Streaming video coding and sla requirements. **IEEE Internet Computing**, v. 13, n. 1, p. 70–75, Janeiro 2009.

[51] ANDY, G.; CLAYPOOL, M. **Trace Analysis Example**. Disponível em: <<http://nile.wpi.edu/NS/analysis.html>>.

[52] SHEN, C.; SCHULZRINNE, H.; NAHUM, E. Principles, systems and applications of ip telecommunications. services and security for next generation networks. Springer-Verlag, Berlin, Heidelberg, p. 149–173, 2008.

[53] ITU-T. **G.1050 Network model for evaluating multimedia transmission performance over Internet Protocol**. Place des Nations, Geneva, Switzerland: International Telecommunication Union, Julho 2016.

## ANEXO 1

```
#Garante que cada vez que for executada uma simulação será utilizada uma  
semente diferente para gerar números randômicos  
  
global defaultRNG  
$defaultRNG seed 0  
  
#Cria a variável n contendo números randômicos uniformemente distribuídos  
entre 0 e 10.  
  
set n [new RandomVariable\Uniform]  
$n set max_ 10  
$n set min_ 0  
  
#Estabelece o tempo em que se dará o início da transmissão da stream de  
vídeo 1:  
  
set Video_Stream_1_Start_Time [$n value]  
$ns at $Video_Stream_1_Start_Time "$Video_Stream_1 start"  
  
#Estabelece o tempo em que se dará o início da transmissão da stream de  
vídeo 2:  
  
set Video_Stream_2_Start_Time [$n value]  
$ns at $Video_Stream_2_Start_Time "$Video_Stream_2 start"
```